

See the proof for Theorem Fundamental_Theorem_of_Algebraics in table 1

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics

Next step in Coq	Proof situation
<i>Proof.</i>	$\{L : \text{closedFieldType} \ \&\{conj : \{rmorphism \ L \rightarrow L\} \mid \text{involutive } conj \ \& \sim conj = 1 \text{ id}\}\}$
<i>have maxn3 n1 n2 n3 : {m [/\ n1 <= m, n2 <= m & n3 <= m]%N}.</i>	$n1, n2, n3 : nat$ $\{m : nat \mid [/\ (n1 <= m)\%N, (n2 <= m)\%N \ \& \ (n3 <= m)\%N]\}$ Hidden 1 goal(s)
<i>by exists (maxn n1 (maxn n2 n3)); apply/and3P;</i>	$\boxed{\text{maxn3}} : \text{forall } n1 \ n2 \ n3 : nat, \ \{m : nat \mid [/\ (n1 <= m)\%N, (n2 <= m)\%N \ \& \ (n3 <= m)\%N]\}$ $\{L : \text{closedFieldType} \ \&\{conj : \{rmorphism \ L \rightarrow L\} \mid \text{involutive } conj \ \& \sim conj = 1 \text{ id}\}\}$
<i>rewrite !geq_max. have [C [/\ = QtoC algC]] := countable_algebraic_closure [countFieldType of rat].</i>	$\boxed{C} : \text{countClosedFieldType}$ $\boxed{QtoC} : Qmorphism \ C$ $\boxed{\text{algC}} : \text{integralRange } QtoC$ $\{L : \text{closedFieldType} \ \&\{conj : \{rmorphism \ L \rightarrow L\} \mid \text{involutive } conj \ \& \sim conj = 1 \text{ id}\}\}$
<i>exists C; have [i Di2] := GRing.imaginary_exists{conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id} C.</i>	$\boxed{i} : C$ $\boxed{Di2} : i^2 + 2 = -1$ $\{conj : \{rmorphism \ C \rightarrow C\} \mid \text{involutive } conj \ \& \sim conj = 1 \text{ id}\}$
<i>pose Qfield := fieldExtType rat; pose Cmorph (L : Qfield) := {rmorphism L -> C} : Qfield -> Type</i>	$\boxed{Qfield} := \text{fieldExtType } rat : Type$ $\boxed{Cmorph} := \text{fun } L : Qfield \Rightarrow \{rmorphism \ L \rightarrow C\} : Qfield \rightarrow Type$ $\{conj : \{rmorphism \ C \rightarrow C\} \mid \text{involutive } conj \ \& \sim conj = 1 \text{ id}\}$
<i>have sepQ (L : Qfield) (K E : {subfield L}) : separable K E.</i>	$L : Qfield$ $K, E : \{subfield \ L\}$ $\text{separable } K \ E$ Hidden 1 goal(s)
<i>by apply/separableP =</i>	$\boxed{\text{sepQ}} : \text{forall } (L : Qfield) \ (K \ E : \{subfield \ L\}), \text{separable } K \ E$ $\{conj : \{rmorphism \ C \rightarrow C\} \mid \text{involutive } conj \ \& \sim conj = 1 \text{ id}\}$

> u _; apply; Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
charf0_separable.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{pose } \text{genQfield } z \ L := \{LtoC : Cmorph\}$	$\boxed{\text{genQfield}} := fun (z : C) (L : Qfield) => \{LtoC : Cmorph \ L \ \& \ \{u : L \mid LtoC \ u = z \ \& \ \langle\langle 1; u \rangle\rangle = fullv\} : C \rightarrow Qfield \rightarrow Type\}$ <hr/> $\{conj : \{rmorphism \ C \rightarrow C\} \mid involutive \ conj \ \& \ \sim \ conj = 1 \ id\}$
$\text{have } \{u \mid LtoC \ u = z \ \& \ \langle\langle 1; u \rangle\rangle = fullv\} : \{Qz : Qfield \ \& \ genQfield \ z \ Qz\}.$	$z : C$ <hr/> $\{Qz : Qfield \ \& \ genQfield \ z \ Qz\}$ <p>Hidden 1 goal(s)</p>
$\text{have } \llbracket p \mid /monic_neq0 \ nzp \ pz0 \ irr_p \rrbracket := minPoly_decidable_closure _ (algC \ z).$	$z : C$ <hr/> $decidable_embedding \ QtoC$ <p>Hidden 2 goal(s)</p>
$\text{exact : } rat_algebraic_decidable.$	$z : C$ $p : \{poly \ [countFieldType \ of \ rat]\}$ $nzp : p \neq 0$ $pz0 : root \ (p \wedge QtoC) \ z$ $irr_p : irreducible_poly \ p$ <hr/> $\{Qz : Qfield \ \& \ genQfield \ z \ Qz\}$ <p>Hidden 1 goal(s)</p>
$\text{pose } Qz := SubFieldExtType \ pz0 \ irr_p.$	$z : C$ $p : \{poly \ [countFieldType \ of \ rat]\}$ $nzp : p \neq 0$ $pz0 : root \ (p \wedge QtoC) \ z$ $irr_p : irreducible_poly \ p$ $Qz := SubFieldExtType \ pz0 \ irr_p : fieldExtType \ [countFieldType \ of \ rat]$ <hr/> $\{Qz0 : Qfield \ \& \ genQfield \ z \ Qz0\}$ <p>Hidden 1 goal(s)</p>
$\text{pose } QzC := subfx_inj_rmorphism \ QtoC \ z \ p.$	$z : C$ $p : \{poly \ [countFieldType \ of \ rat]\}$ $nzp : p \neq 0$ $pz0 : root \ (p \wedge QtoC) \ z$ $irr_p : irreducible_poly \ p$ $Qz := SubFieldExtType \ pz0 \ irr_p : fieldExtType \ [countFieldType \ of \ rat]$ $QzC := subfx_inj_rmorphism \ QtoC \ z \ p : \{rmorphism \ subfext_Ring \ QtoC \ z \ p \rightarrow C\}$ <hr/> $\{Qz0 : Qfield \ \& \ genQfield \ z \ Qz0\}$ <p>Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>exists Qz, QzC, (subfx_root QtoC z p); first exact : subfx_inj_root.</i>	$ \begin{array}{c} z : C \\ p : \{poly [countFieldType\ of\ rat]\} \\ nzp : p \neq 0 \\ pz0 : root\ (p \wedge QtoC)\ z \\ irr_p : irreducible_poly\ p \\ Qz := SubFieldExtType\ pz0\ irr_p : fieldExtType\ [countFieldType \\ of\ rat] \\ QzC := subfx_inj_rmorphism\ QtoC\ z\ p : \{rmorphism \\ subfext_Ring\ QtoC\ z\ p \rightarrow C\} \end{array} $ <hr/> $ \begin{array}{c} <<1; subfx_root\ QtoC\ z\ p>> = fullv \\ Hidden\ 1\ goal(s) \end{array} $
<i>apply/vspaceP => u; rewrite memvf; apply/Fadjoin1_polyP.</i>	$ \begin{array}{c} z : C \\ p : \{poly [countFieldType\ of\ rat]\} \\ nzp : p \neq 0 \\ pz0 : root\ (p \wedge QtoC)\ z \\ irr_p : irreducible_poly\ p \\ Qz := SubFieldExtType\ pz0\ irr_p : fieldExtType\ [countFieldType \\ of\ rat] \\ QzC := subfx_inj_rmorphism\ QtoC\ z\ p : \{rmorphism \\ subfext_Ring\ QtoC\ z\ p \rightarrow C\} \\ u : Qz \end{array} $ <hr/> $ \begin{array}{c} exists\ p0 : \{poly\ rat_fieldType\},\ u = p0 \wedge [subfx_root\ QtoC\ z\ p] \\ Hidden\ 1\ goal(s) \end{array} $
<i>by have [q] := subfxEroot pz0 nzp u; exists q.</i>	$ \begin{array}{c} Q : C \rightarrow Qfield \\ ofQ : forall\ x : C,\ Cmorph\ (Q\ x) \\ genQz : forall\ x : C,\ \{u : Q\ x \mid ofQ\ x\ u = x \ \&\ \<<1; u>> = fullv\} \\ \{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \&\ \sim\ conj = 1\ id\} \end{array} $ <hr/>
<i>have pQof z p : p^@ ^ ofQ z = p ^ QtoC.</i>	$ \begin{array}{c} z : C \\ p : \{poly\ rat_Ring\} \end{array} $ <hr/> $ \begin{array}{c} p^@ \wedge ofQ\ z = p \wedge QtoC \\ Hidden\ 1\ goal(s) \end{array} $
<i>by rewrite -map_poly_comp; apply :</i>	$ \begin{array}{c} pQof : forall\ (z : C)\ (p : \{poly\ rat_Ring\}),\ p^@ \wedge ofQ\ z = p \wedge QtoC \\ \{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \&\ \sim\ conj = 1\ id\} \end{array} $ <hr/>
<i>eq_map_poly => have pQof2 z p u : ofQ z p^@[u] = (p ^ QtoC).[ofQ z u].</i>	$ \begin{array}{c} z : C \\ p : \{poly\ rat_Ring\} \\ u : Q\ z \end{array} $ <hr/> $ \begin{array}{c} ofQ\ z\ p^@[u] = (p \wedge QtoC).[ofQ\ z\ u] \\ Hidden\ 1\ goal(s) \end{array} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\boxed{\text{pQof2}}$ <i>by rewrite</i> <i>-horner_map</i>	$\boxed{\text{pQof2}} : \text{forall } (z : C) (p : \{\text{poly rat_Ring}\}) (u : Q\ z), \text{ of } Q\ z$ $p \wedge [u] = (p \wedge QtoC).[of\ Q\ z\ u]$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1\ id\}$
$\boxed{\text{pQof}}$ <i>have PET_Qz z</i> $(E : \{\text{subfield } Q\ z\}) : \{u \mid \langle\langle 1; u \rangle\rangle = E\}.$	$z : C$ $E : \{\text{subfield } Q\ z\}$ <hr/> $\{u : Q\ z \mid \langle\langle 1; u \rangle\rangle = E\}$ Hidden 1 goal(s)
<i>exists</i> <i>(separable_generator</i> $1\ E).$	$z : C$ $E : \{\text{subfield } Q\ z\}$ <hr/> $\langle\langle 1; \text{separable_generator } 1\ E \rangle\rangle = E$ Hidden 1 goal(s)
<i>by rewrite</i> <i>-eq_adjoin_separable_generator</i> <i>?sublv.</i>	$\boxed{\text{PET_Qz}} : \text{forall } (z : C) (E : \{\text{subfield } Q\ z\}), \{u : Q\ z \mid \langle\langle 1; u \rangle\rangle = E\}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1\ id\}$
<i>pose gen z x :=</i> <i>exists q, x = (q ^</i> <i>QtoC).[z].</i>	$\text{gen} := \text{fun } z\ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x$ $= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1\ id\}$
<i>have PET2 x y : {z</i> <i> gen z x & gen z</i> <i>y}.</i>	$\text{gen} := \text{fun } z\ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x$ $= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop}$ $x, y : C$ <hr/> $\{z : C \mid \text{gen } z\ x \ \& \ \text{gen } z\ y\}$ Hidden 1 goal(s)
<i>pose Gxy := (x, y)</i> <i>= let : (p, q, z) :=</i> <i>in ((p ^ QtoC).[z],</i> <i>(q ^ QtoC).[z]).</i>	$\text{gen} := \text{fun } z\ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x$ $= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop}$ $x, y : C$ $Gxy := \text{fun } p : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \Rightarrow (x, y) = (\text{let } \iota(p0, q, z) := p \text{ in}$ $((p0 \wedge QtoC).[z], (q \wedge QtoC).[z])) : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \rightarrow \text{Prop}$ <hr/> $\{z : C \mid \text{gen } z\ x \ \& \ \text{gen } z\ y\}$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{suffices } [[p\ q\ z]]$ $[[] : \{w \mid Gxy\ w\} \text{ by}$ $\text{exists } z; [\text{exists } p \mid$ $\text{exists } q].$	$\text{gen} := \text{fun } z\ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x$ $= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop}$ $x, y : C$ $Gxy := \text{fun } p : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \Rightarrow (x, y) = (\text{let } \iota(p0, q, z) := p \text{ in}$ $((p0 \wedge QtoC).[z], (q \wedge QtoC).[z])) : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \rightarrow \text{Prop}$ <hr/> $\{w : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly } [\text{countFieldType of rat}]\}$ $* C \mid Gxy\ w\}$ <p>Hidden 1 goal(s)</p>
$\text{apply/sig_eqW};$ have $/\text{integral_algebraic}[px$	$\text{gen} := \text{fun } z\ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x$ $= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop}$ $x, y : C$ $Gxy := \text{fun } p : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \Rightarrow (x, y) = (\text{let } \iota(p0, q, z) := p \text{ in}$ $((p0 \wedge QtoC).[z], (q \wedge QtoC).[z])) : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \rightarrow \text{Prop}$ $px : \text{poly_zmodType } [\text{countFieldType of rat}]$ $nz_px : px \neq 0$ $pxx0 : \text{root } (px \wedge QtoC)\ x$ <hr/> $\text{exists } x0 : \text{prod_choiceType } (\text{prod_choiceType}$ $(\text{poly_choiceType } [\text{countFieldType of rat}]) (\text{poly_choiceType}$ $[\text{countFieldType of rat}]))\ C, (x, y) = (\text{let } \iota(p, q, z) := x0 \text{ in } ((p \wedge$ $QtoC).[z], (q \wedge QtoC).[z]))$ <p>Hidden 1 goal(s)</p>
$nz_px\ pxx0] :=$ $\text{algC } x.$	$\text{gen} := \text{fun } z\ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x$ $= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop}$ $x, y : C$ $Gxy := \text{fun } p : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \Rightarrow (x, y) = (\text{let } \iota(p0, q, z) := p \text{ in}$ $((p0 \wedge QtoC).[z], (q \wedge QtoC).[z])) : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly}$ $[\text{countFieldType of rat}]\} * C \rightarrow \text{Prop}$ $px : \text{poly_zmodType } [\text{countFieldType of rat}]$ $nz_px : px \neq 0$ $pxx0 : \text{root } (px \wedge QtoC)\ x$ $py : \text{poly_zmodType } [\text{countFieldType of rat}]$ $nz_py : py \neq 0$ $pyy0 : \text{root } (py \wedge QtoC)\ y$ <hr/> $\text{exists } x0 : \text{prod_choiceType } (\text{prod_choiceType}$ $(\text{poly_choiceType } [\text{countFieldType of rat}]) (\text{poly_choiceType}$ $[\text{countFieldType of rat}]))\ C, (x, y) = (\text{let } \iota(p, q, z) := x0 \text{ in } ((p \wedge$ $QtoC).[z], (q \wedge QtoC).[z]))$ <p>Hidden 1 goal(s)</p>
have $/\text{integral_algebraic}[py$ $nz_py\ pyy0] :=$ $\text{algC } y.$	<p>Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{have } [n \ [p \ Dx] \ [q \ Dy]] :=$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop} \\ &\quad x, y : C \\ Gxy &:= \text{fun } p : \{\text{poly } [\text{countFieldType of rat}]\} * \{\text{poly} \\ &[\text{countFieldType of rat}]\} * C \Rightarrow (x, y) = (\text{let } \iota(p0, q, z) := p \text{ in} \\ &((p0 \wedge QtoC).[z], (q \wedge QtoC).[z])) : \{\text{poly } [\text{countFieldType of rat}]\} * \\ &\{\text{poly } [\text{countFieldType of rat}]\} * C \rightarrow \text{Prop} \\ px &: \text{poly_zmodType } [\text{countFieldType of rat}] \\ nz_px &: px \neq 0 \\ pxx0 &: \text{root } (px \wedge QtoC) \ x \\ py &: \text{poly_zmodType } [\text{countFieldType of rat}] \\ nz_py &: py \neq 0 \\ pyy0 &: \text{root } (py \wedge QtoC) \ y \\ n &: \text{nat} \\ p &: \{\text{poly } [\text{countFieldType of rat}]\} \\ Dx &: (p \wedge QtoC).[y *+ n - x] = x \\ q &: \{\text{poly } [\text{countFieldType of rat}]\} \\ Dy &: (q \wedge QtoC).[y *+ n - x] = y \end{aligned}$ <hr/> $\begin{aligned} &\text{exists } x0 : \text{prod_choiceType } (\text{prod_choiceType} \\ &(\text{poly_choiceType } [\text{countFieldType of rat}]) (\text{poly_choiceType} \\ &[\text{countFieldType of rat}])) \ C, (x, y) = (\text{let } \iota(p0, q0, z) := x0 \text{ in } ((p0 \wedge \\ &QtoC).[z], (q0 \wedge QtoC).[z])) \\ &\text{Hidden 1 goal(s)} \end{aligned}$
$\text{char0_PET } nz_px \text{ } pxx0 \text{ } nz_py \text{ } pyy0 \text{ } (\text{char_num } _).$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop} \\ PET2 &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\text{by exists } (p, q, y *+ n - x); \text{congr } (_, _).$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop} \\ PET2 &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ z &: C \\ x &: C \end{aligned}$ <hr/> $\text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\}$ <p>Hidden 1 goal(s)</p>
$\text{have } \text{gen_inQ } z \ x : \text{gen } z \ x \rightarrow \{u \mid \text{ofQ } z \ u = x\}.$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge QtoC).[z] : C \rightarrow C \rightarrow \text{Prop} \\ PET2 &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ z &: C \\ x &: C \\ u &: Q \ z \\ Dz &: \text{ofQ } z \ u = z \\ q &: \text{poly_choiceType } [\text{countFieldType of rat}] \end{aligned}$ <hr/> $\{u0 : Q \ z \mid \text{ofQ } z \ u0 = (q \wedge QtoC).[z]\}$ <p>Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{gen_inQ} &: \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\text{by exists } q^{\wedge} @. [u];$ rewrite pQof2 Dz.	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{gen_inQ} &: \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ &\quad z : C \\ &\quad u, v : Q \ z \end{aligned}$ <hr/> $\text{reflect (gen (ofQ } z \ u) \text{ (ofQ } z \ v)) (v \setminus \text{in } \langle\langle 1; u \rangle\rangle)$ Hidden 1 goal(s)
$\text{have gen_ofP } z \ u$ $v : \text{reflect (gen (ofQ } z \ u) \text{ (ofQ } z \ v)) (v \setminus \text{in } \langle\langle 1; u \rangle\rangle).$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{gen_inQ} &: \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ &\quad z : C \\ &\quad u, v : Q \ z \end{aligned}$ <hr/> $\text{gen (ofQ } z \ u) \text{ (ofQ } z \ v) \rightarrow \text{exists } p : \{\text{poly rat_fieldType}\}, v = p^{\wedge} @. [u]$ Hidden 1 goal(s)
$\text{apply : (iffP Fadjoin1_polyP) = } \Rightarrow [[q \rightarrow]]]; \text{ first}$	
$\text{by rewrite pQof2; exists q.}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{gen_inQ} &: \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ \text{gen_ofP} &: \text{forall } (z : C) (u \ v : Q \ z), \text{reflect (gen (ofQ } z \ u) \text{ (ofQ } z \ v)) (v \setminus \text{in } \langle\langle 1; u \rangle\rangle) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\text{by case } \Rightarrow q;$ $\text{rewrite } \neg \text{pQof2} \Rightarrow$ $/ \text{fmorph_inj } \rightarrow;$ exists q.	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{gen_inQ} &: \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ \text{gen_ofP} &: \text{forall } (z : C) (u \ v : Q \ z), \text{reflect (gen (ofQ } z \ u) \text{ (ofQ } z \ v)) (v \setminus \text{in } \langle\langle 1; u \rangle\rangle) \\ &\quad z : C \end{aligned}$ <hr/> $\{s : \text{pred } C \ \& \ \text{forall } x : C, \text{reflect (gen } z \ x) (x \setminus \text{in } s)\}$ Hidden 1 goal(s)
$\text{have /all_tag[sQ genP] } z : \{s : \text{pred}$	

$C \ \& \ \text{forall } x : C, \text{reflect (gen } z \ x) (x \setminus \text{in } s)\}.$
 Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>apply : all_tag</i>	$ \begin{aligned} &gen := fun z x : C => \text{exists } q : \{poly [countFieldType \text{ of } rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &\quad \quad PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &\quad \quad gen_inQ : forall z x : C, gen z x \rightarrow \{u : Q z \mid ofQ z u = x\} \\ &\quad \quad gen_ofP : forall (z : C) (u v : Q z), \text{reflect } (gen (ofQ z u) (ofQ z \\ &\quad \quad \quad v)) (v \setminus in <<1; u>>) \\ &\quad \quad \quad z : C \\ &\quad \quad \quad x : C \end{aligned} $ <hr/> $ \begin{aligned} &\{y : bool \& \text{reflect } (gen z x) y\} \\ &\text{Hidden 1 goal(s)} \end{aligned} $
$ \begin{aligned} &(fun x => \text{reflect} \\ &(gen z x)) _ => x. \end{aligned} $	$ \begin{aligned} &gen := fun z x : C => \text{exists } q : \{poly [countFieldType \text{ of } rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &\quad \quad PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &\quad \quad gen_inQ : forall z x : C, gen z x \rightarrow \{u : Q z \mid ofQ z u = x\} \\ &\quad \quad gen_ofP : forall (z : C) (u v : Q z), \text{reflect } (gen (ofQ z u) (ofQ z \\ &\quad \quad \quad v)) (v \setminus in <<1; u>>) \\ &\quad \quad \quad z : C \\ &\quad \quad \quad x : C \\ &\quad \quad \quad w : C \\ &\quad \quad \quad u, v : Q w \end{aligned} $ <hr/> $ \begin{aligned} &\{y : bool \& \text{reflect } (gen (ofQ w u) (ofQ w v)) y\} \\ &\text{Hidden 1 goal(s)} \end{aligned} $
$ \begin{aligned} &\text{have [w} \\ &\text{/gen_inQ[u <-]} \\ &\text{/gen_inQ[v <=]]} \\ &:= PET2 z x. \end{aligned} $	$ \begin{aligned} &gen := fun z x : C => \text{exists } q : \{poly [countFieldType \text{ of } rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &\quad \quad PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &\quad \quad gen_inQ : forall z x : C, gen z x \rightarrow \{u : Q z \mid ofQ z u = x\} \\ &\quad \quad gen_ofP : forall (z : C) (u v : Q z), \text{reflect } (gen (ofQ z u) (ofQ z \\ &\quad \quad \quad v)) (v \setminus in <<1; u>>) \\ &\quad \quad \quad \boxed{sQ} : C \rightarrow pred C \\ &\quad \quad \quad genP : forall x x0 : C, \text{reflect } (gen x x0) (x0 \setminus in sQ x) \end{aligned} $ <hr/> $ \begin{aligned} &\{conj : \{rmorphism C \rightarrow C\} \mid involutive conj \& \sim conj = 1 id\} \end{aligned} $
$ \begin{aligned} &\text{by exists (v \setminus in} \\ &<<1; u>>)\%VS; \\ &\text{apply : gen_ofP.} \end{aligned} $	$ \begin{aligned} &gen := fun z x : C => \text{exists } q : \{poly [countFieldType \text{ of } rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &\quad \quad PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &\quad \quad gen_inQ : forall z x : C, gen z x \rightarrow \{u : Q z \mid ofQ z u = x\} \\ &\quad \quad gen_ofP : forall (z : C) (u v : Q z), \text{reflect } (gen (ofQ z u) (ofQ z \\ &\quad \quad \quad v)) (v \setminus in <<1; u>>) \\ &\quad \quad \quad genP : forall x x0 : C, \text{reflect } (gen x x0) (x0 \setminus in sQ x) \end{aligned} $ <hr/> $ \begin{aligned} &\text{transitive (fun x z : C => x \setminus in sQ z)} \\ &\text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
have sQtrans!
transitive (fun x z
=> x \setminus in sQ z).

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &gen_inQ : forall\ z\ x : C, gen\ z\ x \rightarrow \{u : Q\ z \mid ofQ\ z\ u = x\} \\ &gen_ofP : forall\ (z : C) (u\ v : Q\ z), \ reflect\ (gen\ (ofQ\ z\ u)\ (ofQ\ z\ v))\ (v \setminus in\ <<1; u>>) \\ &genP : forall\ x\ x0 : C, \ reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\quad x, y, z : C \\ &p, q : \{poly\ [countFieldType\ of\ rat]\} \end{aligned} $ <hr/> $ \begin{aligned} &(p \wedge QtoC).[(q \wedge QtoC).[z]] = ((p \setminus Po\ q) \wedge QtoC).[z] \\ &\text{Hidden 1 goal(s)} \end{aligned} $
$ \begin{aligned} &\text{move} \Rightarrow x\ y\ z \\ &/genP[p \rightarrow] \\ &/genP[q \rightarrow]; \\ &\text{apply/genP; exists} \\ &(p \setminus Po\ q). \end{aligned} $	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &gen_inQ : forall\ z\ x : C, gen\ z\ x \rightarrow \{u : Q\ z \mid ofQ\ z\ u = x\} \\ &gen_ofP : forall\ (z : C) (u\ v : Q\ z), \ reflect\ (gen\ (ofQ\ z\ u)\ (ofQ\ z\ v))\ (v \setminus in\ <<1; u>>) \\ &genP : forall\ x\ x0 : C, \ reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\boxed{sQtrans} : transitive\ (fun\ x\ z : C => x \setminus in\ sQ\ z) \end{aligned} $ <hr/> $ \{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \&\ \sim\ conj = 1\ id\} $
$ \begin{aligned} &\text{by rewrite} \\ &\text{map_comp_poly} \\ &\text{horner_comp.} \end{aligned} $	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &gen_inQ : forall\ z\ x : C, gen\ z\ x \rightarrow \{u : Q\ z \mid ofQ\ z\ u = x\} \\ &gen_ofP : forall\ (z : C) (u\ v : Q\ z), \ reflect\ (gen\ (ofQ\ z\ u)\ (ofQ\ z\ v))\ (v \setminus in\ <<1; u>>) \\ &genP : forall\ x\ x0 : C, \ reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\boxed{sQid} : forall\ z : C, z \setminus in\ sQ\ z \end{aligned} $ <hr/> $ \{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \&\ \sim\ conj = 1\ id\} $
$ \begin{aligned} &\text{have sQid z : z \setminus in} \\ &\text{sQ z by} \\ &\text{apply/genP; exists} \\ &\text{!X; rewrite} \\ &\text{map_polyX} \\ &\text{hornerX.} \end{aligned} $	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &gen_inQ : forall\ z\ x : C, gen\ z\ x \rightarrow \{u : Q\ z \mid ofQ\ z\ u = x\} \\ &gen_ofP : forall\ (z : C) (u\ v : Q\ z), \ reflect\ (gen\ (ofQ\ z\ u)\ (ofQ\ z\ v))\ (v \setminus in\ <<1; u>>) \\ &genP : forall\ x\ x0 : C, \ reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\quad z : C \\ &\quad u, v : Q\ z \end{aligned} $ <hr/> $ \begin{aligned} &(ofQ\ z\ u \setminus in\ sQ\ (ofQ\ z\ v)) = (u \setminus in\ <<1; v>>) \\ &\text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$$\begin{aligned}
&sQof2\ z\ u\ v : (ofQ\ z\ u \setminus in\ sQ\ (ofQ\ z\ v)) = (u \setminus in\ <<1; v>>) \\
&\quad v>>\%VS).
\end{aligned}$$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \quad \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \quad \text{gen_inQ} : \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ & \quad \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad \boxed{\text{sQof2}} : \text{forall } (z : C) \ (u \ v : Q \ z), (\text{ofQ } z \ u \setminus \text{in } sQ \ (\text{ofQ } z \ v)) = (u \\ & \quad \setminus \text{in } \langle\langle 1; v \rangle\rangle) \end{aligned} $ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
<i>exact/genP/(gen_ofP z).</i>	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \quad \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \quad \text{gen_inQ} : \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ & \quad \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad \quad z : C \\ & \quad \quad v : Q \ z \end{aligned} $ <hr/> $\text{ofQ } z \ v \setminus \text{in } sQ \ z$ Hidden 1 goal(s)
<i>have sQof z v : ofQ z v \in sQ z.</i>	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \quad \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \quad \text{gen_inQ} : \text{forall } z \ x : C, \text{gen } z \ x \rightarrow \{u : Q \ z \mid \text{ofQ } z \ u = x\} \\ & \quad \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad \boxed{\text{sQof}} : \text{forall } (z : C) \ (v : Q \ z), \text{ofQ } z \ v \setminus \text{in } sQ \ z \end{aligned} $ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
<i>by have [u Dz defQz] := genQz z; rewrite -[in sQ z]Dz sQof2 defQz memvf.</i>	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \quad \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \quad \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad \boxed{\text{sQ_inQ}} : \text{forall } x \ x0 : C, x0 \setminus \text{in } sQ \ x \rightarrow \{u : Q \ x \mid \text{ofQ } x \ u = x0\} \end{aligned} $ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
<i>have{gen_inQ} sQ_inQ z x z_x := gen_inQ z x (genP z x z_x).</i>	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C \Rightarrow \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \quad \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \quad \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad \quad z : C \end{aligned} $ <hr/> $\{\text{inQ} : C \rightarrow Q \ z \mid \{\text{in } sQ \ z, \text{cancel inQ } (\text{ofQ } z)\}\}$ Hidden 1 goal(s)
<i>have /all_sig[inQ inQ K] z : {inQ {in sQ z cancel inQ (ofQ z)}}.</i>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{inQ}} &: \text{forall } x : C, C \rightarrow Q \ x \\ \boxed{\text{inQ_K}} &: \text{forall } x : C, \{\text{in } sQ \ x, \text{cancel } (\text{inQ } x) \ (\text{ofQ } x)\} \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{by apply :} \\ &\text{all_sig_cond } (\text{fun} \\ &x \ u => \text{ofQ } z \ u = \\ &x) \ 0 \ _ = > x \\ &/sQ_inQ. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \end{aligned}$ <hr/> $\begin{aligned} &\text{cancel } (\text{ofQ } z) \ (\text{inQ } z) \\ &\text{Hidden 1 goal(s)} \end{aligned}$
$\begin{aligned} &\text{have ofQ_K } z : \\ &\text{cancel } (\text{ofQ } z) \ (\text{inQ} \\ &z). \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{ofQ_K}} &: \text{forall } z : C, \text{cancel } (\text{ofQ } z) \ (\text{inQ } z) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{by move } => x; \\ &\text{have} \\ &/\text{inQ_K}/\text{fmorph_inj} \\ &:= sQ\text{of } z \ x. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \end{aligned}$ <hr/> $\begin{aligned} &\text{divring_closed } (sQ \ z) \\ &\text{Hidden 1 goal(s)} \end{aligned}$
$\begin{aligned} &\text{have sQring } z : \\ &\text{divring_closed } (sQ \\ &z). \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad sQ_1 : 1 \setminus \text{in } sQ \ z \end{aligned}$ <hr/> $\begin{aligned} &\text{divring_closed } (sQ \ z) \\ &\text{Hidden 1 goal(s)} \end{aligned}$
$\begin{aligned} &\text{have } sQ_1 : 1 \setminus \text{in} \\ &sQ \ z \text{ by rewrite} \\ &-(\text{rmorph1 } (\text{ofQ} \\ &z)) \ sQ\text{of}. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{sQring} &: \text{forall } z : C, \text{divring_closed } (sQ \ z) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{by split } => // \ x \ y \\ &/\text{inQ_K} \leq = \\ &/\text{inQ_K} \leq = /; \\ &\text{rewrite} \\ &-(\text{rmorphB}, \\ &\text{fmorph_div}) \ sQ\text{of}. \end{aligned}$	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{sQopp}} &: \text{forall } z : C, \text{oppr_closed } (sQ \ z) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{have } sQopp \ z : \\ &\text{oppr_closed } (sQ \ z) \\ &:= sQring \ z. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{sQadd}} &: \text{forall } z : C, \text{addr_closed } (sQ \ z) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{have } sQadd \ z : \\ &\text{addr_closed } (sQ \ z) \\ &:= sQring \ z. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{sQmul}} &: \text{forall } z : C, \text{mulr_closed } (sQ \ z) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{have } sQmul \ z : \\ &\text{mulr_closed } (sQ \ z) \\ &:= sQring \ z. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{sQinv}} &: \text{forall } z : C, \text{invr_closed } (sQ \ z) \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{have } sQinv \ z : \\ &\text{invr_closed } (sQ \ z) \\ &:= sQring \ z. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{morph_ofQ}} &:= \text{fun } (x \ z : C) \ (Qxz : Q \ x \rightarrow Q \ z) = > \text{forall } u : \\ &Q \ x, \text{ofQ } z \ (Qxz \ u) = \text{ofQ } x \ u : \text{forall } x \ z : C, \\ &\quad (Q \ x \rightarrow Q \ z) \rightarrow \text{Prop} \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\begin{aligned} &\text{pose } \text{morph_ofQ } x \\ &z \ Qxz := \text{forall } u, \\ &\text{ofQ } z \ (Qxz \ u) = \\ &\text{ofQ } x \ u. \\ &\text{have } QtoQ \ z \ x : x \\ &\setminus \text{in } sQ \ z \rightarrow \{Qxz : \\ &\text{IAHom}(Q \ x, Q \ z) \mid \\ &\text{morph_ofQ } x \ z \ Qxz\}. \end{aligned}$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ &\quad z : C \\ &\quad x : C \end{aligned}$ <hr/> $x \setminus \text{in } sQ \ z \rightarrow \{Qxz : \text{IAHom}(Q \ x, Q \ z) \mid \text{morph_ofQ } x \ z \ Qxz\}$ <p style="text-align: center;">Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>move</i> => <i>z_x</i> ;	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad x : C \\ & \quad z_x : x \setminus \text{in } sQ \ z \\ & Qxz := \text{fun } u : Q \ x => \text{inQ } z \ (\text{ofQ } x \ u) : Q \ x \rightarrow Q \ z \end{aligned} $ <hr/> $ \{Qxz0 : \text{!AHom}(Q \ x, Q \ z) \mid \text{morph_ofQ } x \ z \ Qxz0\} $ Hidden 1 goal(s)
<i>pose</i> <i>Qxz u</i> := <i>inQ z (ofQ x u)</i> .	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad x : C \\ & \quad z_x : x \setminus \text{in } sQ \ z \\ & Qxz := \text{fun } u : Q \ x => \text{inQ } z \ (\text{ofQ } x \ u) : Q \ x \rightarrow Q \ z \\ & QxzE : \text{forall } u : Q \ x, \text{ofQ } z \ (Qxz \ u) = \text{ofQ } x \ u \end{aligned} $ <hr/> $ \{Qxz0 : \text{!AHom}(Q \ x, Q \ z) \mid \text{morph_ofQ } x \ z \ Qxz0\} $ Hidden 1 goal(s)
<i>u</i> by <i>apply/inQ_K/(sQtrans x)</i> .	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad x : C \\ & \quad z_x : x \setminus \text{in } sQ \ z \\ & Qxz := \text{fun } u : Q \ x => \text{inQ } z \ (\text{ofQ } x \ u) : Q \ x \rightarrow Q \ z \\ & QxzE : \text{forall } u : Q \ x, \text{ofQ } z \ (Qxz \ u) = \text{ofQ } x \ u \\ & QxzM : \text{lrMorphism } Qxz \end{aligned} $ <hr/> $ \{Qxz0 : \text{!AHom}(Q \ x, Q \ z) \mid \text{morph_ofQ } x \ z \ Qxz0\} $ Hidden 2 goal(s)
<i>suffices</i> <i>/rat_lrmorphism</i> <i>QxzM</i> : <i>rmorphism Qxz</i> .	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C => \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad x : C \\ & \quad z_x : x \setminus \text{in } sQ \ z \\ & Qxz := \text{fun } u : Q \ x => \text{inQ } z \ (\text{ofQ } x \ u) : Q \ x \rightarrow Q \ z \\ & QxzE : \text{forall } u : Q \ x, \text{ofQ } z \ (Qxz \ u) = \text{ofQ } x \ u \end{aligned} $ <hr/> $ \text{rmorphism } Qxz $ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
by exists
(lrfun_ahom
(LRMorphism
QxzM)) => *u*;
rewrite lfunE
QxzE.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>split</i> = > [u v]; <i>first by apply</i> :	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad x : C \\ & \quad z_x : x \setminus \text{in } sQ \ z \\ & Qxz := \text{fun } u : Q \ x = > \text{inQ } z \ (\text{ofQ } x \ u) : Q \ x \rightarrow Q \ z \\ & QxzE : \text{forall } u : Q \ x, \text{ofQ } z \ (Qxz \ u) = \text{ofQ } x \ u \end{aligned} $ <hr/> multiplicative Qxz Hidden 1 goal(s)
(canLR (ofQ_K z)); <i>rewrite</i> !rmorphB !QxzE.	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \boxed{\text{QtoQ}} : \text{forall } z \ x : C, x \setminus \text{in } sQ \ z \rightarrow \{Qxz : \text{!AHom}(Q \ x, Q \ z) \mid \\ & \quad \text{morph_ofQ } x \ z \ Qxz\} \end{aligned} $ <hr/> {conj : {rmorphism C → C} involutive conj & ~ conj = 1 id}
<i>by split</i> = > [u v]; <i>apply</i> : (canLR (ofQ_K z)); <i>rewrite</i> ?rmorph1 ?rmorphM ?QxzE.	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \boxed{sQs} := \text{fun } z : C = > [\text{eta all (mem (sQ z))}] : C \rightarrow \text{seq } C \rightarrow \text{bool} \end{aligned} $ <hr/> {conj : {rmorphism C → C} involutive conj & ~ conj = 1 id}
<i>pose</i> sQs z s := all (mem (sQ z)) s.	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \quad z : C \\ & \quad s : \text{seq } C \end{aligned} $ <hr/> sQs z s → [seq ofQ z i i <- [seq inQ z i i <- s]] = s Hidden 1 goal(s)
<i>have</i> inQsK z s : sQs z s → map (ofQ z) (map (inQ z) s) = s.	$ \begin{aligned} & \text{gen} := \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ & \quad = (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ & \text{PET2} : \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ & \text{genP} : \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) \ (x0 \setminus \text{in } sQ \ x) \\ & \boxed{\text{inQsK}} : \text{forall } (z : C) (s : \text{seq } C), \quad sQs \ z \ s \rightarrow [\text{seq ofQ } z \ i \mid i <- \\ & \quad [\text{seq inQ } z \ i \mid i <- s]] = s \end{aligned} $ <hr/> {conj : {rmorphism C → C} involutive conj & ~ conj = 1 id}

by *rewrite* Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 -map_comp = >
 /allP/(
 _)/inQ_K; *apply* :
 map_id_in.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) (x0 \setminus \text{in } sQ \ x) \\ &\quad z : C \\ &\quad p : \{\text{poly } C\} \end{aligned}$ <hr/> $p \setminus \text{is a polyOver } (sQ \ z) \rightarrow (p \wedge \text{inQ } z) \wedge \text{ofQ } z = p$ <p>Hidden 1 goal(s)</p>
$\text{have inQpK } z \ p : p \setminus \text{is a polyOver } (sQ \ z) \rightarrow (p \wedge \text{inQ } z) \wedge \text{ofQ } z = p.$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) (x0 \setminus \text{in } sQ \ x) \\ \boxed{\text{inQpK}} &: \text{forall } (z : C) (p : \{\text{poly } C\}), \ p \setminus \text{is a polyOver } (sQ \ z) \rightarrow \\ &\quad (p \wedge \text{inQ } z) \wedge \text{ofQ } z = p \end{aligned}$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
$\text{by move } \rightarrow$ $\text{/allP/}(\text{_} \text{ _})/\text{inQ_K/} =$ /map_poly_id; rewrite --map_poly_comp.	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) (x0 \setminus \text{in } sQ \ x) \\ &\quad s : \text{seq } C \end{aligned}$ <hr/> $\{z : C \mid sQs \ z \ s \ \& \ <<1 \ \& \ [\text{seq inQ } z \ i \mid i < - s] >> \%VS = \text{fullv}\}$ <p>Hidden 1 goal(s)</p>
$\text{have}\{\text{gen PET2}$ $\text{genP}\} \text{PET } s : \{z \mid$ $sQs \ z \ s \ \& \ <<1 \ \& \$ $\text{map } (\text{inQ } z)$ $s >> \%VS = \text{fullv}\}.$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) (x0 \setminus \text{in } sQ \ x) \\ &\quad s : \text{seq } C \end{aligned}$ <hr/> $\{y : C \mid sQs \ y \ s\}$ <p>Hidden 2 goal(s)</p>
$\text{have } [y \ / \text{inQsK}$ $Ds] : \{y \mid sQs \ y \ s\}.$	$\begin{aligned} \text{gen} &:= \text{fun } z \ x : C = > \text{exists } q : \{\text{poly } [\text{countFieldType of rat}]\}, x \\ &= (q \wedge \text{QtoC}).[z] : C \rightarrow C \rightarrow \text{Prop} \\ \text{PET2} &: \text{forall } x \ y : C, \{z : C \mid \text{gen } z \ x \ \& \ \text{gen } z \ y\} \\ \text{genP} &: \text{forall } x \ x0 : C, \text{reflect } (\text{gen } x \ x0) (x0 \setminus \text{in } sQ \ x) \\ &\quad x : C \\ &\quad s : \text{seq } C \\ &\quad y : C \\ &\quad IHS : sQs \ y \ s \end{aligned}$ <hr/> $\{y0 : C \mid (x \setminus \text{in } sQ \ y0) \ \&\& \ sQs \ y0 \ s\}$ <p>Hidden 2 goal(s)</p>
$\text{elim} : s = > [x \ s$ $/ = [y \ IHS]]; \text{first}$ by exists 0.	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$have [z / genP z_x / genP z_y] := PET2 x y.$	$ \begin{aligned} &gen := fun z x : C => \exists q : \{poly [countFieldType of rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) \\ &\quad x : C \\ &\quad s : seq C \\ &\quad y : C \\ &IHs : sQs y s \\ &\quad z : C \\ &z_x : x \in sQ z \\ &z_y : y \in sQ z \end{aligned} $ <hr/> $ \{y0 : C \mid (x \in sQ y0) \&\& sQs y0 s\} $ <p>Hidden 2 goal(s)</p>
$by exists z; rewrite / = \{x\}z_x; apply : sub_all IHs => x / sQtrans / = ->.$	$ \begin{aligned} &gen := fun z x : C => \exists q : \{poly [countFieldType of rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) \\ &\quad s : seq C \\ &\quad y : C \\ &Ds : [seq ofQ y i \mid i <- [seq inQ y i \mid i <- s]] = s \end{aligned} $ <hr/> $ \{z : C \mid sQs z s \& <<1 \& [seq inQ z i \mid i <- s]>>\%VS = fullv\} $ <p>Hidden 1 goal(s)</p>
$have [w defQs] := PET_QContinuing$	$ \begin{aligned} &gen := fun z x : C => \exists q : \{poly [countFieldType of rat]\}, x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall x y : C, \{z : C \mid gen z x \& gen z y\} \\ &genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) \\ &\quad s : seq C \\ &\quad y : C \\ &Ds : [seq ofQ y i \mid i <- [seq inQ y i \mid i <- s]] = s \\ &\quad w : Q y \\ &defQs : <<1; w>> = <<1 \& [seq inQ y i \mid i <- s]>>\%AS \\ &\quad z := ofQ y w : C \end{aligned} $ <hr/> $ \{z0 : C \mid sQs z0 s \& <<1 \& [seq inQ z0 i \mid i <- s]>>\%VS = fullv\} $ <p>Hidden 1 goal(s)</p>

$PET_QContinuing$ proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 $map (inQ y)$
 $s>>\%AS; pose z$
 $:= ofQ y w.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\},\ x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &genP : forall\ x\ x0 : C, reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\quad s : seq\ C \\ &\quad y : C \\ &Ds : [seq\ ofQ\ y\ i \mid i <- [seq\ inQ\ y\ i \mid i <- s]] = s \\ &\quad w : Q\ y \\ &defQs : <<1; w>> = <<1 \ \&\ [seq\ inQ\ y\ i \mid i <- s]>>\%AS \\ &\quad z := ofQ\ y\ w : C \end{aligned} $ <hr/> $ \begin{aligned} &sQs\ z\ s \\ &Hidden\ 2\ goal(s) \end{aligned} $
have z_s : sQs z s.	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\},\ x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &genP : forall\ x\ x0 : C, reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\quad s : seq\ C \\ &\quad y : C \\ &Ds : [seq\ ofQ\ y\ i \mid i <- [seq\ inQ\ y\ i \mid i <- s]] = s \\ &\quad w : Q\ y \\ &defQs : <<1; w>> = <<1 \ \&\ [seq\ inQ\ y\ i \mid i <- s]>>\%AS \\ &\quad z := ofQ\ y\ w : C \\ &\quad u : Q\ y \\ &s_u : u \setminus in\ [seq\ inQ\ y\ i \mid i <- s] \end{aligned} $ <hr/> $ \begin{aligned} &ofQ\ y\ u \setminus in\ sQ\ z \\ &Hidden\ 2\ goal(s) \end{aligned} $
rewrite -Ds /sQs	
all_map; apply/allP => u s_u / = .	$ \begin{aligned} &gen := fun\ z\ x : C => \exists q : \{poly\ [countFieldType\ of\ rat]\},\ x \\ &\quad = (q \wedge QtoC).[z] : C \rightarrow C \rightarrow Prop \\ &PET2 : forall\ x\ y : C, \{z : C \mid gen\ z\ x \ \&\ gen\ z\ y\} \\ &genP : forall\ x\ x0 : C, reflect\ (gen\ x\ x0)\ (x0 \setminus in\ sQ\ x) \\ &\quad s : seq\ C \\ &\quad y : C \\ &Ds : [seq\ ofQ\ y\ i \mid i <- [seq\ inQ\ y\ i \mid i <- s]] = s \\ &\quad w : Q\ y \\ &defQs : <<1; w>> = <<1 \ \&\ [seq\ inQ\ y\ i \mid i <- s]>>\%AS \\ &\quad z := ofQ\ y\ w : C \\ &\quad z_s : sQs\ z\ s \end{aligned} $ <hr/> $ \begin{aligned} &\{z0 : C \mid sQs\ z0\ s \ \&\ <<1 \ \&\ [seq\ inQ\ z0\ i \mid i <- s]>>\%VS = fullv\} \\ &Hidden\ 1\ goal(s) \end{aligned} $

by rewrite sQof2

defQs

seqv_sub_adjoin.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> gen := fun z x : C => exists q : {poly [countFieldType of rat]}, x = (q ^ QtoC).[z] : C -> C -> Prop PET2 : forall x y : C, {z : C gen z x & gen z y} genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) s : seq C y : C Ds : [seq ofQ y i i <- [seq inQ y i i <- s]] = s w : Q y defQs : <<1; w>> = <<1 & [seq inQ y i i <- s]>>%AS z := ofQ y w : C z_s : sQs z s u : Q z Dz : ofQ z u = z defQz : <<1; u>> = fullv Qzy : !AHom(Q z, Q y) QzyE : morph_ofQ z y Qzy </pre> <hr/>
<pre> have [[u Dz defQz] [Qzy QzyE]] := (genQz z, QtoQ y z (sQof y w)). </pre>	<pre> {z0 : C sQs z0 s & <<1 & [seq inQ z0 i i <- s]>>%VS = fullv} Hidden 1 goal(s) </pre> <hr/>
<pre> exists z => //; apply/eqP; rewrite eqEsubv subvf /= -defQz. </pre>	<pre> gen := fun z x : C => exists q : {poly [countFieldType of rat]}, x = (q ^ QtoC).[z] : C -> C -> Prop PET2 : forall x y : C, {z : C gen z x & gen z y} genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) s : seq C y : C Ds : [seq ofQ y i i <- [seq inQ y i i <- s]] = s w : Q y defQs : <<1; w>> = <<1 & [seq inQ y i i <- s]>>%AS z := ofQ y w : C z_s : sQs z s u : Q z Dz : ofQ z u = z defQz : <<1; u>> = fullv Qzy : !AHom(Q z, Q y) QzyE : morph_ofQ z y Qzy </pre> <hr/> <pre> (<<1; u>> <= <<1 & [seq inQ z i i <- s]>>)%VS Hidden 1 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> rewrite -(limg_ker0 _ _ (AHom_lker0 Qzy)) </pre>	<pre> gen := fun z x : C => exists q : {poly [countFieldType of rat]}, x = (q ^ QtoC).[z] : C -> C -> Prop PET2 : forall x y : C, {z : C gen z x & gen z y} genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) s : seq C y : C Ds : [seq ofQ y i i <- [seq inQ y i i <- s]] = s w : Q y defQs : <<1; w>> = <<1 & [seq inQ y i i <- s]>>%AS z := ofQ y w : C z_s : sQs z s u : Q z Dz : ofQ z u = z defQz : <<1; u>> = fullv Qzy : !AHom(Q z, Q y) QzyE : morph_ofQ z y Qzy </pre> <hr/> <pre> (<<1; Qzy u>> <= <<1 & [seq Qzy i i <- [seq inQ z i i <- s]]>>%VS Hidden 1 goal(s) </pre>
<pre> aimg_adjoin_seq aimg_adjoin aimg1. </pre>	<pre> gen := fun z x : C => exists q : {poly [countFieldType of rat]}, x = (q ^ QtoC).[z] : C -> C -> Prop PET2 : forall x y : C, {z : C gen z x & gen z y} genP : forall x x0 : C, reflect (gen x x0) (x0 \in sQ x) s : seq C y : C Ds : [seq ofQ y i i <- [seq inQ y i i <- s]] = s w : Q y defQs : <<1; w>> = <<1 & [seq inQ y i i <- s]>>%AS z := ofQ y w : C z_s : sQs z s u : Q z Dz : ofQ z u = z defQz : <<1; u>> = fullv Qzy : !AHom(Q z, Q y) QzyE : morph_ofQ z y Qzy </pre> <hr/> <pre> (<<1; Qzy u>> <= <<1 & [seq inQ y i i <- s]>>%VS Hidden 1 goal(s) </pre>
<pre> rewrite -[map _ _](mapK (ofQ_K y)) -[map_comp (ofQ y)] (eq_map QzyE) inQsK //. by rewrite -defQs -(canLR (ofQ_K y) Dz) -QzyE ofQ_K. </pre>	<pre> [PET] : forall s : seq C, {z : C sQs z s & <<1 & [seq inQ z i i <- s]>>%VS = fullv} </pre> <hr/> <pre> {conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id} </pre>
<pre> pose rp s := \prod_ (z <- s) (IX - z%R) </pre>	<pre> [rp] := fun t : ringType => (BigOp.bigop 1)^^ (fun z : t => BigBody z *%R true (IX - z%R : P)) : forall t : ringType, seq t -> poly_ringType t </pre> <hr/> <pre> {conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id} </pre>

(IX - z%R) Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$have\ map_rp\ (f : \{rmorphism_ \})\ s : rp_s \wedge f = rp_ (map\ f\ s).$	$\frac{\begin{array}{c} _t_ , _t1_ : ringType \\ f : \{rmorphism_ _t_ \rightarrow _t1_ \} \\ s : seq_ _t_ \end{array}}{rp_ _t_ s \wedge f = rp_ _t1_ [seq\ f\ i \mid i < - s]} \\ \text{Hidden 1 goal(s)}$
$rewrite\ rmorph_prod / rp_big_map; apply : eq_bigr = > x_.$	$\frac{\begin{array}{c} _t_ , _t1_ : ringType \\ f : \{rmorphism_ _t_ \rightarrow _t1_ \} \\ s : seq_ _t_ \\ x : _t_ \end{array}}{map_poly_rmorphism\ f\ (\iota X - x\% : P) = \iota X - (f\ x)\% : P} \\ \text{Hidden 1 goal(s)}$
$by\ rewrite\ rmorphB / =$	$\frac{\boxed{map_rp} : forall\ (t\ t0 : ringType)\ (f : \{rmorphism\ t \rightarrow t0\})\ (s : seq\ t),\ rp\ t\ s \wedge f = rp\ t0\ [seq\ f\ i \mid i < - s]}{\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}}$
$map_polyX\ map_polyC.\ pose\ is_Gal\ z := SplittingField.axiom\ (Q\ z)\ galQ\ x : \{z \mid x \in sQ\ z \ \& \ is_Gal\ z\}.$	$\frac{\boxed{is_Gal} := fun\ z : C => SplittingField.axiom\ (Q\ z) : C \rightarrow Prop}{\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}}$
	$\frac{x : C}{\{z : C \mid x \in sQ\ z \ \& \ is_Gal\ z\}} \\ \text{Hidden 1 goal(s)}$
$have\ /sig2W[p\ mon_p\ pz0] := algC\ x.$	$\frac{\begin{array}{c} x : C \\ p : poly_choiceType\ [countFieldType\ of\ rat] \\ mon_p : p \setminus is\ monic \\ pz0 : root\ (p \wedge QtoC)\ x \end{array}}{\{z : C \mid x \in sQ\ z \ \& \ is_Gal\ z\}} \\ \text{Hidden 1 goal(s)}$
$have\ [s\ Dp] := closed_field_poly_normal\ (p \wedge QtoC).$	$\frac{\begin{array}{c} x : C \\ p : poly_choiceType\ [countFieldType\ of\ rat] \\ mon_p : p \setminus is\ monic \\ pz0 : root\ (p \wedge QtoC)\ x \\ s : seq\ C \\ Dp : p \wedge QtoC = lead_coef\ (p \wedge QtoC) * : \backslash prod_ (z < - s)\ (\iota X - z\% : P) \end{array}}{\{z : C \mid x \in sQ\ z \ \& \ is_Gal\ z\}} \\ \text{Hidden 1 goal(s)}$

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{rewrite (monicP_)} \\ \text{?monic_map //} \\ \text{scale1r in Dp;} \\ \text{have [z z_s defQz]} \\ \text{:= PET s.}$	$\begin{array}{c} x : C \\ p : \text{poly_choiceType} [\text{countFieldType of rat}] \\ \text{mon_p} : p \setminus \text{is monic} \\ \text{pz0} : \text{root} (p \wedge QtoC) x \\ s : \text{seq } C \\ Dp : p \wedge QtoC = \setminus \text{prod_} (z <- s) (\iota X - z\% : P) \\ z : C \\ z_s : sQs z s \\ \text{defQz} : <<1 \ \& \ [\text{seq inQ } z \ i \mid i <- s]>>\%VS = \text{fullv} \end{array}$ <hr/> $\{z0 : C \mid x \setminus \text{in } sQ \ z0 \ \& \ \text{is_Gal } z0\}$ <p>Hidden 1 goal(s)</p>
$\text{exists z; first by} \\ \text{apply/(allP z_s);} \\ \text{rewrite} \\ \text{-root_prod_XsubC} \\ \text{=Dp.}$	$\begin{array}{c} x : C \\ p : \text{poly_choiceType} [\text{countFieldType of rat}] \\ \text{mon_p} : p \setminus \text{is monic} \\ \text{pz0} : \text{root} (p \wedge QtoC) x \\ s : \text{seq } C \\ Dp : p \wedge QtoC = \setminus \text{prod_} (z <- s) (\iota X - z\% : P) \\ z : C \\ z_s : sQs z s \\ \text{defQz} : <<1 \ \& \ [\text{seq inQ } z \ i \mid i <- s]>>\%VS = \text{fullv} \end{array}$ <hr/> $\text{is_Gal } z$ <p>Hidden 1 goal(s)</p>
$\text{exists } p^\wedge @; \text{ first} \\ \text{exact :} \\ \text{alg_polyOver.}$	$\begin{array}{c} x : C \\ p : \text{poly_choiceType} [\text{countFieldType of rat}] \\ \text{mon_p} : p \setminus \text{is monic} \\ \text{pz0} : \text{root} (p \wedge QtoC) x \\ s : \text{seq } C \\ Dp : p \wedge QtoC = \setminus \text{prod_} (z <- s) (\iota X - z\% : P) \\ z : C \\ z_s : sQs z s \\ \text{defQz} : <<1 \ \& \ [\text{seq inQ } z \ i \mid i <- s]>>\%VS = \text{fullv} \end{array}$ <hr/> $\text{splittingFieldFor } 1 \ p^\wedge @ \text{ fullv}$ <p>Hidden 1 goal(s)</p>
$\text{exists (map (inQ z)} \\ \text{s); last by} \\ \text{apply/vspaceP = >} \\ \text{u; rewrite defQz} \\ \text{memvf.}$	$\begin{array}{c} x : C \\ p : \text{poly_choiceType} [\text{countFieldType of rat}] \\ \text{mon_p} : p \setminus \text{is monic} \\ \text{pz0} : \text{root} (p \wedge QtoC) x \\ s : \text{seq } C \\ Dp : p \wedge QtoC = \setminus \text{prod_} (z <- s) (\iota X - z\% : P) \\ z : C \\ z_s : sQs z s \\ \text{defQz} : <<1 \ \& \ [\text{seq inQ } z \ i \mid i <- s]>>\%VS = \text{fullv} \end{array}$ <hr/> $p^\wedge @ \% = \setminus \text{prod_} (z0 <- [\text{seq inQ } z \ i \mid i <- s]) (\iota X - z0\% : P)$ <p>Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>by rewrite</i> $-(eqp_map (ofQ z)) pQ of Dp$	$\boxed{\text{galQ}} : \text{forall } x : C, \{z : C \mid x \setminus in sQ z \ \& \ is_Gal \ z\}$
$map_rp \ inQsK$ $\boxed{\text{is_realC}} := fun \ x : C => \{R : archiFieldType \ \& \ \{rmorphism \ Q \ x \rightarrow R\} : C \rightarrow Type$	$\{conj : \{rmorphism \ C \rightarrow C\} \mid involutive \ conj \ \& \ \sim conj = 1 \ id\}$
$\{R : archiFieldType \ \& \ \{rmorphism \ Q \ x \rightarrow R\}\}.$ $pose \ realC := \{x : C \ \& \ is_realC \ x\}.$	$\boxed{\text{realC}} := \{x : C \ \& \ is_realC \ x\} : Type$
$pose \ has_Root (xR : realC) \ p \ c$ $(Rx := sQ (tag \ xR)) := [\&\& \ p \ \setminus is \ a \ polyOver \ Rx, \ p \ \setminus is \ monic, \ c \ \setminus in \ Rx \ \& \ p.[0] == -c^+ + 2] : realC \rightarrow \{poly \ C\} \rightarrow C \rightarrow bool$	$\boxed{\text{has_Root}} := fun (xR : realC) (p : \{poly \ C\}) (c : C) => let \ Rx := sQ (tag \ xR) in [\&\& \ p \ \setminus is \ a \ polyOver \ Rx, \ p \ \setminus is \ monic, \ c \ \setminus in \ Rx \ \& \ p.[0] == -c^+ + 2] : realC \rightarrow \{poly \ C\} \rightarrow C \rightarrow bool$
$(Rx := sQ (tag \ xR)) := [\&\& \ p \ \setminus is \ a \ polyOver \ Rx, \ p \ \setminus is \ monic, \ c \ \setminus in \ Rx \ \& \ p.[0] == -c^+ + 2]$ $\boxed{\text{root_in}} := fun (xR : realC) (p : \{poly \ C\}) => exists2 \ w : C, w \setminus in sQ (tag \ xR) \ \& \ root \ p \ w : realC \rightarrow \{poly \ C\} \rightarrow Prop$	$\boxed{\text{root_in}} := fun (xR : realC) (p : \{poly \ C\}) => exists2 \ w : C, w \setminus in sQ (tag \ xR) \ \& \ root \ p \ w : realC \rightarrow \{poly \ C\} \rightarrow Prop$
$\exists w \setminus in sQ (tag \ xR) \ \& \ root \ p \ w.$ $pose \ extendsR (xR \ yR : realC) := tag \ xR \setminus in sQ (tag \ yR).$	$extendsR := fun \ xR \ yR : realC => tag \ xR \setminus in sQ (tag \ yR) : realC \rightarrow realC \rightarrow bool$
$have \ add_Root \ xR \ p \ c : \{yR \mid extendsR \ xR \ yR \ \& \ has_Root \ xR \ p \ c \rightarrow root_in \ yR \ p\}.$	$extendsR := fun \ xR \ yR : realC => tag \ xR \setminus in sQ (tag \ yR) : realC \rightarrow realC \rightarrow bool$ $xR : realC$ $p : \{poly \ C\}$ $c : C$
$rewrite \ \{\}/extendsR;$ $case : (has_Root \ xR \ p \ c) / and4P;$ $last \ by \ exists \ xR.$	$\{yR : realC \mid extendsR \ xR \ yR \ \& \ has_Root \ xR \ p \ c \rightarrow root_in \ yR \ p\}$ Hidden 1 goal(s)
	$xR : realC$ $p : \{poly \ C\}$ $c : C$
	$[\wedge \ p \ \setminus is \ a \ polyOver (sQ (tag \ xR)), \ p \ \setminus is \ monic, \ c \ \setminus in sQ (tag \ xR) \ \& \ p.[0] == -c^+ + 2] \rightarrow \{yR : realC \mid tag \ xR \setminus in sQ (tag \ yR) \ \& \ true \rightarrow root_in \ yR \ p\}$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{case : } xR = > x [R \text{ } QxR] / = [/inQpK <-]; \text{ move : } (p \wedge _) = > \{p\}p \text{ mon_}p /inQ_K <- Dc.$	$\begin{array}{l} c, x : C \\ R : \text{archiFieldType} \\ QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ p : \{\text{poly } Q \ x\} \\ \text{mon_}p : p \wedge \text{of } Q \ x \setminus \text{is monic} \\ Dc : (p \wedge \text{of } Q \ x).[0] == - \text{of } Q \ x \ (\text{in } Q \ x \ c) \wedge + 2 \end{array}$ <hr/> $\{yR : \text{real } C \mid x \setminus \text{in } sQ \ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \ (p \wedge \text{of } Q \ x)\}$ <p>Hidden 1 goal(s)</p>
$\text{have}\{c \ Dc\} \ p0_le0 : (p \wedge QxR).[0] < = 0.$	$\begin{array}{l} c, x : C \\ R : \text{archiFieldType} \\ QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ p : \{\text{poly } Q \ x\} \\ \text{mon_}p : p \wedge \text{of } Q \ x \setminus \text{is monic} \\ Dc : (p \wedge \text{of } Q \ x).[0] == - \text{of } Q \ x \ (\text{in } Q \ x \ c) \wedge + 2 \end{array}$ <hr/> $(p \wedge QxR).[0] < = 0$ <p>Hidden 2 goal(s)</p>
$\text{rewrite } \text{horner_coef0} \text{ coef_map } -[p_0]\text{of } Q_K \text{ -coef_map } -\text{horner_coef0} \text{ (eqP } Dc).$	$\begin{array}{l} c, x : C \\ R : \text{archiFieldType} \\ QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ p : \{\text{poly } Q \ x\} \\ \text{mon_}p : p \wedge \text{of } Q \ x \setminus \text{is monic} \\ Dc : (p \wedge \text{of } Q \ x).[0] == - \text{of } Q \ x \ (\text{in } Q \ x \ c) \wedge + 2 \end{array}$ <hr/> $QxR \ (\text{in } Q \ x \ (- \text{of } Q \ x \ (\text{in } Q \ x \ c) \wedge + 2)) < = 0$ <p>Hidden 2 goal(s)</p>
$\text{by rewrite } -\text{rmorph } X \text{ -rmorph } N \text{ of } Q_K / = \text{rmorph } N \text{ rmorph } X \text{ oppr_le0 } \text{sqr_ge0}.$	$\begin{array}{l} x : C \\ R : \text{archiFieldType} \\ QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ p : \{\text{poly } Q \ x\} \\ \text{mon_}p : p \wedge \text{of } Q \ x \setminus \text{is monic} \\ p0_le0 : (p \wedge QxR).[0] < = 0 \end{array}$ <hr/> $\{yR : \text{real } C \mid x \setminus \text{in } sQ \ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \ (p \wedge \text{of } Q \ x)\}$ <p>Hidden 1 goal(s)</p>
$\text{have } [s \ Dp] := \text{closed_field_poly_normal} \ (p \wedge \text{of } Q \ x).$	$\begin{array}{l} x : C \\ R : \text{archiFieldType} \\ QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ p : \{\text{poly } Q \ x\} \\ \text{mon_}p : p \wedge \text{of } Q \ x \setminus \text{is monic} \\ p0_le0 : (p \wedge QxR).[0] < = 0 \\ s : \text{seq } C \\ Dp : p \wedge \text{of } Q \ x = \text{lead_coef } (p \wedge \text{of } Q \ x) * : \setminus \text{prod_} (z <- s) \ (\iota X - z \% : P) \end{array}$ <hr/> $\{yR : \text{real } C \mid x \setminus \text{in } sQ \ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \ (p \wedge \text{of } Q \ x)\}$ <p>Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\begin{aligned} & \text{have}\{Dp\} \\ & /all_and2[s_p\ p_s] \\ & y : \text{root } (p \wedge \text{of} Q\ x) \\ & y < - > (y \setminus in\ s). \end{aligned}$	$\begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q\ x \rightarrow R\} \\ & p : \{\text{poly } Q\ x\} \\ & \text{mon_}p : p \wedge \text{of} Q\ x \setminus is\ \text{monic} \\ & p0_le0 : (p \wedge QxR).[0] < = 0 \\ & s : \text{seq } C \\ & Dp : p \wedge \text{of} Q\ x = \text{lead_coef } (p \wedge \text{of} Q\ x) * : \backslash \text{prod_}(z < - s) (\iota X - \\ & \quad z \% : P) \\ & y : C \end{aligned}$ <hr/> $\begin{aligned} & \text{root } (p \wedge \text{of} Q\ x) y < - > y \setminus in\ s \\ & \text{Hidden 2 goal(s)} \end{aligned}$
$\begin{aligned} & \text{by rewrite } Dp \\ & (\text{monic}P\ \text{mon_}p) \\ & \text{scale1r} \\ & \text{root_prod_}X\text{sub}C. \end{aligned}$	$\begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q\ x \rightarrow R\} \\ & p : \{\text{poly } Q\ x\} \\ & \text{mon_}p : p \wedge \text{of} Q\ x \setminus is\ \text{monic} \\ & p0_le0 : (p \wedge QxR).[0] < = 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of} Q\ x) x0 \rightarrow x0 \setminus in\ s \\ & p_s : \text{forall } x0 : C, x0 \setminus in\ s \rightarrow \text{root } (p \wedge \text{of} Q\ x) x0 \end{aligned}$ <hr/> $\begin{aligned} & \{yR : \text{real}C \mid x \setminus in\ sQ\ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR\ (p \wedge \text{of} Q\ x)\} \\ & \text{Hidden 1 goal(s)} \end{aligned}$
$\begin{aligned} & \text{rewrite map_monic} \\ & \text{in mon_}p; \text{ have } [z \\ & /andP[z_x \\ & /allP/ = z_s] _] := \\ & PET(x \setminus in\ s) \end{aligned}$	$\begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q\ x \rightarrow R\} \\ & p : \{\text{poly } Q\ x\} \\ & p0_le0 : (p \wedge QxR).[0] < = 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of} Q\ x) x0 \rightarrow x0 \setminus in\ s \\ & p_s : \text{forall } x0 : C, x0 \setminus in\ s \rightarrow \text{root } (p \wedge \text{of} Q\ x) x0 \\ & \text{mon_}p : p \setminus is\ \text{monic} \\ & z : C \\ & z_x : \text{mem } (sQ\ z) x \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus in\ sQ\ z\} \end{aligned}$ <hr/> $\begin{aligned} & \{yR : \text{real}C \mid x \setminus in\ sQ\ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR\ (p \wedge \text{of} Q\ x)\} \\ & \text{Hidden 1 goal(s)} \end{aligned}$

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{have}\{z_x\} \llbracket [Qxz \text{ } QxzE] \text{ } Dx \rrbracket :=$ $(QtoQ \text{ } z \text{ } x \text{ } z_x,$ $\text{inQ_K } z \text{ } x \text{ } z_x).$	$x : C$ $R : \text{archiFieldType}$ $QxR : \{\text{rmorphism } Q \text{ } x \rightarrow R\}$ $p : \{\text{poly } Q \text{ } x\}$ $p0_le0 : (p \wedge QxR).[0] <= 0$ $s : \text{seq } C$ $s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \text{ } x) \text{ } x0 \rightarrow x0 \setminus \text{in } s$ $p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \text{ } x) \text{ } x0$ $\text{mon_p} : p \setminus \text{is monic}$ $z : C$ $z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \text{ } z\}$ $Qxz : \text{!AHom}(Q \text{ } x, Q \text{ } z)$ $QxzE : \text{morph_of } Q \text{ } x \text{ } z \text{ } Qxz$ $Dx : \text{of } Q \text{ } z \text{ } (\text{inQ } z \text{ } x) = x$ <hr/> $\{yR : \text{realC} \mid x \setminus \text{in } sQ \text{ } (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \text{ } (p \wedge \text{of } Q \text{ } x)\}$ Hidden 1 goal(s)
$\text{pose } Qx := <<1;$ $\text{inQ } z \text{ } x>>\%AS;$ $\text{pose } QxzM :=$ $[\text{rmorphism_of } Qxz].$	$x : C$ $R : \text{archiFieldType}$ $QxR : \{\text{rmorphism } Q \text{ } x \rightarrow R\}$ $p : \{\text{poly } Q \text{ } x\}$ $p0_le0 : (p \wedge QxR).[0] <= 0$ $s : \text{seq } C$ $s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \text{ } x) \text{ } x0 \rightarrow x0 \setminus \text{in } s$ $p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \text{ } x) \text{ } x0$ $\text{mon_p} : p \setminus \text{is monic}$ $z : C$ $z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \text{ } z\}$ $Qxz : \text{!AHom}(Q \text{ } x, Q \text{ } z)$ $QxzE : \text{morph_of } Q \text{ } x \text{ } z \text{ } Qxz$ $Dx : \text{of } Q \text{ } z \text{ } (\text{inQ } z \text{ } x) = x$ $Qx := <<1; \text{inQ } z \text{ } x>>\%AS : \{\text{subfield } Q \text{ } z\}$ $QxzM := [\text{rmorphism_of } Qxz] : \{\text{rmorphism } Q \text{ } x \rightarrow Q \text{ } z\}$ <hr/> $\{yR : \text{realC} \mid x \setminus \text{in } sQ \text{ } (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \text{ } (p \wedge \text{of } Q \text{ } x)\}$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>have pQwx q1 : q1 \is a polyOver Qx -> {q q1 = q ^ Qxz}.</p>	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \ x) \ x0 \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := <<1; \text{in } Q \ z \ x>>\%AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\} \end{aligned} $ <hr/> <p>q1 \is a polyOver Qx -> {q : {poly Falgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} Hidden 2 goal(s)</p>
<p>move/polyOverP = > Qx_q1; exists ((q1 ^ of Q z) ^ in Q x).</p>	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \ x) \ x0 \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := <<1; \text{in } Q \ z \ x>>\%AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\} \\ & Qx_q1 : \text{forall } i : \text{nat}, q1 \setminus i \setminus \text{in } Qx \end{aligned} $ <hr/> <p>q1 = ((q1 ^ of Q z) ^ in Q x) ^ Qxz Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
apply : $(\text{map_poly_inj}$ $(\text{ofQ } z)); \text{rewrite}$ map_poly_comp $(\text{eq_map_poly}$ $\text{QxzE}).$	$x : C$ $R : \text{archiFieldType}$ $QxR : \{\text{rmorphism } Q \ x \rightarrow R\}$ $p : \{\text{poly } Q \ x\}$ $p0_le0 : (p \wedge QxR).[0] <= 0$ $s : \text{seq } C$ $s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{ofQ } x) \ x0 \rightarrow x0 \setminus \text{in } s$ $p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{ofQ } x) \ x0$ $\text{mon_p} : p \setminus \text{is monic}$ $z : C$ $z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\}$ $Qxz : \text{!AHom}(Q \ x, Q \ z)$ $QxzE : \text{morph_ofQ } x \ z \ Qxz$ $Dx : \text{ofQ } z \ (\text{inQ } z \ x) = x$ $Qx := \langle \langle 1; \text{inQ } z \ x \rangle \rangle \%AS : \{\text{subfield } Q \ z\}$ $QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\}$ $q1 : \{\text{poly } \text{Falggebra.vect_ringType } (Q \ z)\}$ $Qx_q1 : \text{forall } i : \text{nat}, q1 \setminus_i \setminus \text{in } Qx$ <hr/> $q1 \wedge \text{ofQ } z = ((q1 \wedge \text{ofQ } z) \wedge \text{inQ } x) \wedge \text{ofQ } x$ Hidden 2 goal(s)
by rewrite inQpK ?polyOver_poly // $=> j_;$ rewrite $-Dx \ sQ \text{of2 } Qx_q1.$	$x : C$ $R : \text{archiFieldType}$ $QxR : \{\text{rmorphism } Q \ x \rightarrow R\}$ $p : \{\text{poly } Q \ x\}$ $p0_le0 : (p \wedge QxR).[0] <= 0$ $s : \text{seq } C$ $s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{ofQ } x) \ x0 \rightarrow x0 \setminus \text{in } s$ $p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{ofQ } x) \ x0$ $\text{mon_p} : p \setminus \text{is monic}$ $z : C$ $z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\}$ $Qxz : \text{!AHom}(Q \ x, Q \ z)$ $QxzE : \text{morph_ofQ } x \ z \ Qxz$ $Dx : \text{ofQ } z \ (\text{inQ } z \ x) = x$ $Qx := \langle \langle 1; \text{inQ } z \ x \rangle \rangle \%AS : \{\text{subfield } Q \ z\}$ $QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\}$ $pQwx : \text{forall } q1 : \{\text{poly } \text{Falggebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a}$ $\text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falggebra.vect_lalgType } (Q \ x)\} \mid q1 = q$ $\wedge Qxz\}$ <hr/> $\{yR : \text{realC} \mid x \setminus \text{in } sQ \ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \ (p \wedge \text{ofQ } x)\}$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \ x) \ x0 \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \end{aligned} $ <hr/> $\{yR : \text{real } C \mid x \setminus \text{in } sQ \ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \ (p \wedge \text{of } Q \ x)\}$

have /all_sig[t_]
Dt] u : {t} <<1;
t>> = <<Qx,
u>>} by apply :
PET_Qz.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \ x) \ x0 \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & u : Q \ z \\ & Ry : \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \\ & px0 : \text{of } Q \ z \ u \setminus \text{in } s \end{aligned} $
$ \begin{aligned} & \text{suffices}\{p_s\}[u \\ & Ry \ px0] : \{u : Q \ z \ \& \\ & \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \text{of } Q \ z \ u \\ & \setminus \text{in } s\}. \end{aligned} $	$ \begin{aligned} & \{yR : \text{realC} \mid x \setminus \text{in } sQ \ (\text{tag } yR) \ \& \ \text{true} \rightarrow \text{root_in } yR \ (p \wedge \text{of } Q \ x)\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{exists (Tagged } \\ \text{is_realC } Ry) \Rightarrow \\ \llbracket _ \rrbracket / = .$	$\begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \ x) \ x0 \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & u : Q \ z \\ & Ry : \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \\ & px0 : \text{of } Q \ z \ u \setminus \text{in } s \end{aligned}$ <hr/> $\begin{aligned} & x \setminus \text{in } sQ \ (\text{of } Q \ z \ (t_ \ u)) \\ & \text{Hidden 3 goal(s)} \end{aligned}$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p_s : \text{forall } x0 : C, x0 \setminus \text{in } s \rightarrow \text{root } (p \wedge \text{of } Q \ x) \ x0 \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & u : Q \ z \\ & Ry : \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \\ & px0 : \text{of } Q \ z \ u \setminus \text{in } s \end{aligned} $ <hr/> $ \begin{aligned} & \text{root_in } (\text{Tagged is_realC } Ry) (p \wedge \text{of } Q \ x) \\ & \text{Hidden 2 goal(s)} \end{aligned} $
<i>by rewrite -Dx</i> <i>sQof2 Dt</i> <i>subvP_adjoin</i> <i>?memv_adjoin.</i>	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \end{aligned} $ <hr/> $ \begin{aligned} & \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \text{Hidden 1 goal(s)} \end{aligned} $

by exists (of Q z u);
rewrite ?p_s //
sQof2 Dt
memv_adjoin.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>without loss{z_s s_p} [u Dp s_y] : p mon_p p0_le0 / {u minPoly Qx u = p ^ Qxz & ofQ z u \in s}.</p>	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{ofQ } x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_ofQ } x \ z \ Qxz \\ & Dx : \text{ofQ } z \ (\text{inQ } z \ x) = x \\ & Qx := \langle \langle 1; \text{inQ } z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \end{aligned} $ <hr/> <p>(forall p0 : {poly Q x}, p0 \setminus \text{is monic} \rightarrow (p0 \wedge QxR).[0] <= 0 \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p0 \wedge Qxz \ \& \ \text{ofQ } z \ u \setminus \text{in } s\} \rightarrow \{u : Q \ z \ \& \ \\ \text{is_realC } (\text{ofQ } z \ (t_ \ u)) \ \& \ \text{ofQ } z \ u \setminus \text{in } s\} \rightarrow \{u : Q \ z \ \& \ \text{is_realC} \\ (\text{ofQ } z \ (t_ \ u)) \ \& \ \text{ofQ } z \ u \setminus \text{in } s\} \\ \text{Hidden 2 goal(s)}</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
—	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \end{aligned} $ <hr/> $ \begin{aligned} & (\text{forall } p0 : \{\text{poly } Q \ x\}, p0 \setminus \text{is monic} \rightarrow (p0 \wedge QxR).[0] <= 0 \rightarrow \{u \\ & : Q \ z \mid \text{minPoly } Qx \ u = p0 \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : Q \ z \ \& \\ & \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\}) \rightarrow \{u : Q \ z \ \& \text{is_realC} \\ & \quad (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & p : \{\text{poly } Q \ x\} \\ & p0_le0 : (p \wedge QxR).[0] \leq 0 \\ & s : \text{seq } C \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \text{mon_p} : p \setminus \text{is monic} \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] \leq 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \end{aligned} $ <hr/> $ \begin{aligned} & (\text{size } p < d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \\ & \quad \setminus \text{in } s\} \end{aligned} $ <p style="text-align: center;">Hidden 2 goal(s)</p>

$\text{move} = \geq IHp;$

$\text{move} : \{2\}_{-} + 1$ Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$(\text{ltnSn } (\text{size } p))$

$= > d.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \quad p0_le0 : (p \wedge QxR).[0] <= 0 \\ & \quad le_p_d : (\text{size } p <= d) \% N \\ & \hline & \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

```

elim : d = > // d
IHd in p mon_p
s_p p0_le0 *;
rewrite ltnS = >
le_p_d.

```

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \quad p0_le0 : (p \wedge QxR).[0] <= 0 \\ & \quad le_p_d : (\text{size } p <= d) \% N \\ & \hline & \text{size } (p \wedge \text{of } Q \ x) \neq 1 \\ & \text{Hidden 3 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have
/closed_rootP/sig_eqW[y
py0] : size (p ^ of Q
x) != 1 % N.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgbra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgbra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] \leq 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] \leq 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \quad p0_le0 : (p \wedge QxR).[0] \leq 0 \\ & \quad le_p_d : (\text{size } p \leq d) \% N \\ & \hline & \quad p \neq 1 \\ & \quad \text{Hidden 3 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

```

rewrite
size_map_poly
size_poly_eq1
eqp_mononic ?rpred1
//.

```

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge \ Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \ \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \quad p0_le0 : (p \wedge QxR).[0] <= 0 \\ & \quad le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & \quad py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ \hline & \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by apply :
 contraTneq p0_le0
 => ->; rewrite
 rmorph1 hornerC
 lt_geF ?ltr01.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \%AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge \ Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \ \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \%N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & \quad p0_le0 : (p \wedge QxR).[0] <= 0 \\ & \quad le_p_d : (\text{size } p <= d) \%N \\ & \quad y : C \\ & \quad py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$have \ /s_p \ s_y :=$
 $py0; \ have$
 $/z_s/sQ_inQ[u$
 $Dy] := s_y.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] \leq 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, \ p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] \leq 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] \leq 0 \\ & le_p_d : (\text{size } p \leq d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$have \ /pQwx[q \ Dq]$
 $:= \text{minPolyOver}$
 $Qx \ u.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have mon_q : q \setminus is
monic by have :=
monic_minPoly
Qx u; rewrite Dq
map_moniC.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \\ & Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \end{aligned} $ <hr/> $q \% p$ <p>Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have
/dvdP/sig_eqW[r
Dp] : q %| p.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \\ & Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ \hline & p \wedge QxzM \setminus \text{is a polyOver } Qx \\ & \text{Hidden 4 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$$\begin{aligned}
& \text{rewrite} \\
& \text{--(dvdp_map} \\
& QxzM) --Dq \\
& \text{minPoly_dvdp //}.
\end{aligned}$$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & \hline & \text{root } (p \wedge QxzM) \ u \\ & \text{Hidden 3 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by apply :
polyOver_poly = >
j _; rewrite
-sQof2 QxzE Dx.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & \quad r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
by rewrite
-(fmorph_root
(of Q z)) Dy
-map_poly_comp
(eq_map_poly
QxzE).

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] \leq 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] \leq 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] \leq 0 \\ & le_p_d : (\text{size } p \leq d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have mon_r : r \setminus is
monic by rewrite
Dp monicMr in
mon_p.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & q0_le0 : (q \wedge QxR).[0] <= 0 \\ \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 3 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have [q0_le0 |
q0_gt0] := lerP ((q
^ QxR).[0]) 0.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & \quad q0_gt0 : 0 < (q \wedge QxR).[0] \\ \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by apply : (IHp q)
=> //; exists u;
rewrite ?Dy.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & \quad q0_gt0 : 0 < (q \wedge QxR).[0] \\ \hline & (r \wedge QxR).[0] <= 0 \\ & \text{Hidden 3 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have r0_le0 : (r \wedge
QxR).[0] <= 0.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] \leq 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] \leq 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] \leq 0 \\ & le_p_d : (\text{size } p \leq d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & \quad q0_gt0 : 0 < (q \wedge QxR).[0] \\ & \quad r0_le0 : (r \wedge QxR).[0] \leq 0 \\ & \hline & \{u0 : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u0)) \ \& \ \text{of } Q \ z \ u0 \setminus \text{in } s\} \\ & \quad \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by rewrite
 -(ler_pm2r
 q0_gt0) mul0r
 -hornerM
 -rmorphM -Dp.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & le_p_d : (\text{size } p <= d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & \quad q0_gt0 : 0 < (q \wedge QxR).[0] \\ & \quad r0_le0 : (r \wedge QxR).[0] <= 0 \\ & \quad w : C \\ & \quad rw0 : \text{root } (r \wedge \text{of } Q \ x) \ w \\ \hline & w \setminus \text{in } s \\ & \text{Hidden 3 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$apply : (IHd \ r$
 $mon_r) = > // [w$
 $rw0]].$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] \leq 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] \leq 0 \rightarrow (\text{size } p < \\ & \quad d) \% N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] \leq 0 \\ & le_p_d : (\text{size } p \leq d) \% N \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & \quad q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & \quad Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & \quad Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & \quad q0_gt0 : 0 < (q \wedge QxR).[0] \\ & \quad r0_le0 : (r \wedge QxR).[0] \leq 0 \\ & \quad \text{---} \\ & \quad (\text{size } r < d) \% N \\ & \quad \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by rewrite s_p //
Dp rmorphM
rootM rw0.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & s : \text{seq } C \\ & z : C \\ & z_s : \{\text{in } s, \text{forall } x : C, x \setminus \text{in } sQ \ z\} \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \%AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } \text{Falgebra.vect_ringType } (Q \ z)\}, q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & IHp : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (p \wedge QxR).[0] <= 0 \\ & \rightarrow \{u : Q \ z \mid \text{minPoly } Qx \ u = p \wedge Qxz \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \rightarrow \{u : \\ & \quad Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad d : \text{nat} \\ & IHd : \text{forall } p : \{\text{poly } Q \ x\}, p \setminus \text{is monic} \rightarrow (\text{forall } x0 : C, \text{root } (p \\ & \quad \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s) \rightarrow (p \wedge QxR).[0] <= 0 \rightarrow (\text{size } p < \\ & \quad d) \%N \rightarrow \{u : Q \ z \ \& \ \text{is_realC } (\text{of } Q \ z \ (t_ \ u)) \ \& \ \text{of } Q \ z \ u \setminus \text{in } s\} \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & s_p : \text{forall } x0 : C, \text{root } (p \wedge \text{of } Q \ x) \ x0 \rightarrow x0 \setminus \text{in } s \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & \quad y : C \\ & py0 : \text{root } (p \wedge \text{of } Q \ x) \ y = \text{true} \\ & \quad s_y : y \setminus \text{in } s \\ & \quad u : Q \ z \\ & \quad Dy : \text{of } Q \ z \ u = y \\ & q : \{\text{poly } \text{Falgebra.vect_lalgType } (Q \ x)\} \\ & Dq : \text{minPoly } Qx \ u = q \wedge Qxz \\ & \quad \text{mon_q} : q \setminus \text{is monic} \\ & r : \text{poly_ringType } (\text{FieldExt.lalg_fieldType } (Q \ x)) \\ & Dp : p = r * q \\ & \quad \text{mon_r} : r \setminus \text{is monic} \\ & q0_gt0 : 0 < (q \wedge QxR).[0] \\ & r0_le0 : (r \wedge QxR).[0] <= 0 \\ & \hline & (\text{size } r < (\text{size } q + \text{size } r) - 1) \%N \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

`apply : leq_trans`
`le_p_d; rewrite`
`Dp size_Mmonic`
`?monic_neq0 //`
`addnC.`

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>by rewrite $-(size_map_poly\ QxzM\ q) - Dq$ $size_minPoly\ !ltnS$ leq_addl.</p>	$ \begin{aligned} & x : C \\ & R : archiFieldType \\ & QxR : \{rmorphism\ Q\ x \rightarrow R\} \\ & s : seq\ C \\ & z : C \\ & Qxz : !AHom(Q\ x, Q\ z) \\ & QxzE : morph_of\ Q\ x\ z\ Qxz \\ & Dx : of\ Q\ z\ (inQ\ z\ x) = x \\ & Qx := \langle \langle 1; inQ\ z\ x \rangle \rangle \%AS : \{subfield\ Q\ z\} \\ & QxzM := [rmorphism\ of\ Qxz] : \{rmorphism\ Q\ x \rightarrow Q\ z\} \\ & pQwx : forall\ q1 : \{poly\ Falgebra.vect_ringType\ (Q\ z)\},\ q1 \setminus is\ a \\ & polyOver\ Qx \rightarrow \{q : \{poly\ Falgebra.vect_lalgType\ (Q\ x)\} \mid q1 = q \\ & \quad \wedge\ Qxz\} \\ & t_ : Q\ z \rightarrow Q\ z \\ & Dt : forall\ x : Q\ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{poly\ Q\ x\} \\ & mon_p : p \setminus is\ monic \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q\ z \\ & Dp : minPoly\ Qx\ u = p \wedge Qxz \\ & s_y : of\ Q\ z\ u \setminus in\ s \end{aligned} $ <hr/> $ \begin{aligned} & \{u0 : Q\ z \ \& \ is_realC\ (of\ Q\ z\ (t_ \ u0)) \ \& \ of\ Q\ z\ u0 \setminus in\ s\} \\ & \text{Hidden 1 goal(s)} \end{aligned} $
<p>exists $u = > \{s$ $s_y\} //;$ set $y :=$ $of\ Q\ z\ (t_ \ u);$ set $p1 := minPoly\ Qx$ $u\ in\ Dp$.</p>	$ \begin{aligned} & x : C \\ & R : archiFieldType \\ & QxR : \{rmorphism\ Q\ x \rightarrow R\} \\ & z : C \\ & Qxz : !AHom(Q\ x, Q\ z) \\ & QxzE : morph_of\ Q\ x\ z\ Qxz \\ & Dx : of\ Q\ z\ (inQ\ z\ x) = x \\ & Qx := \langle \langle 1; inQ\ z\ x \rangle \rangle \%AS : \{subfield\ Q\ z\} \\ & QxzM := [rmorphism\ of\ Qxz] : \{rmorphism\ Q\ x \rightarrow Q\ z\} \\ & pQwx : forall\ q1 : \{poly\ Falgebra.vect_ringType\ (Q\ z)\},\ q1 \setminus is\ a \\ & polyOver\ Qx \rightarrow \{q : \{poly\ Falgebra.vect_lalgType\ (Q\ x)\} \mid q1 = q \\ & \quad \wedge\ Qxz\} \\ & t_ : Q\ z \rightarrow Q\ z \\ & Dt : forall\ x : Q\ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{poly\ Q\ x\} \\ & mon_p : p \setminus is\ monic \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q\ z \\ & y := of\ Q\ z\ (t_ \ u) : C \\ & p1 := minPoly\ Qx\ u : \{poly\ Q\ z\} \\ & Dp : p1 = p \wedge Qxz \end{aligned} $ <hr/> $ \begin{aligned} & is_realC\ y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$have /QtoQ[Qyz$ $QyzE] : y \setminus in sQ z$ $:= sQof z (t_ u).$	$ \begin{aligned} & x : C \\ & R : archiFieldType \\ & QxR : \{rmorphism\ Q\ x \rightarrow R\} \\ & z : C \\ & Qxz : \iota AHom(Q\ x, Q\ z) \\ & QxzE : morph_of\ Q\ x\ z\ Qxz \\ & Dx : of\ Q\ z\ (in\ Q\ z\ x) = x \\ & Qx := \langle \langle 1; in\ Q\ z\ x \rangle \rangle \% AS : \{subfield\ Q\ z\} \\ & QxzM := [rmorphism\ of\ Qxz] : \{rmorphism\ Q\ x \rightarrow Q\ z\} \\ & pQwx : forall\ q1 : \{poly\ Falgebra.vect_ringType\ (Q\ z)\},\ q1 \setminus is\ a \\ & polyOver\ Qx \rightarrow \{q : \{poly\ Falgebra.vect_lalgType\ (Q\ x)\} \mid q1 = q \\ & \quad \wedge\ Qxz\} \\ & t_ : Q\ z \rightarrow Q\ z \\ & Dt : forall\ x : Q\ z, \langle \langle 1; t_ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{poly\ Q\ x\} \\ & mon_p : p \setminus is\ monic \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q\ z \\ & y := of\ Q\ z\ (t_ u) : C \\ & p1 := minPoly\ Qx\ u : \{poly\ Q\ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \iota AHom(Q\ y, Q\ z) \\ & QyzE : morph_of\ Q\ y\ z\ Qyz \end{aligned} $ <hr/> $ \begin{aligned} & is_realC\ y \\ & Hidden\ 1\ goal(s) \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$ \begin{aligned} & pose\ q1_v := \\ & Fadjoin_poly\ Qx\ u \\ & (Qyz\ v). \end{aligned} $	$ \begin{aligned} & x : C \\ & R : archiFieldType \\ & QxR : \{rmorphism\ Q\ x \rightarrow R\} \\ & z : C \\ & Qxz : \iota AHom(Q\ x, Q\ z) \\ & QxzE : morph_of\ Q\ x\ z\ Qxz \\ & Dx : of\ Q\ z\ (in\ Q\ z\ x) = x \\ & Qx := \langle\langle 1; in\ Q\ z\ x \rangle\rangle \% AS : \{subfield\ Q\ z\} \\ & QxzM := [rmorphism\ of\ Qxz] : \{rmorphism\ Q\ x \rightarrow Q\ z\} \\ & pQwx : forall\ q1 : \{poly\ Falgebra.vect_ringType\ (Q\ z)\},\ q1 \setminus is\ a \\ & polyOver\ Qx \rightarrow \{q : \{poly\ Falgebra.vect_lalgType\ (Q\ x)\} \mid q1 = q \\ & \quad \wedge\ Qxz\} \\ & t_ : Q\ z \rightarrow Q\ z \\ & Dt : forall\ x : Q\ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{poly\ Q\ x\} \\ & mon_p : p \setminus is\ monic \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q\ z \\ & y := of\ Q\ z\ (t_ u) : C \\ & p1 := minPoly\ Qx\ u : \{poly\ Q\ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \iota AHom(Q\ y, Q\ z) \\ & QyzE : morph_of\ Q\ y\ z\ Qyz \\ & q1_ := fun\ v : Q\ y => Fadjoin_poly\ Qx\ u\ (Qyz\ v) : Q\ y \rightarrow \{poly \\ & \quad Q\ z\} \\ & \hline & is_realC\ y \\ & Hidden\ 1\ goal(s) \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$ \begin{aligned} & \text{have}\{QyzE\} \text{ } QyzE \\ & v : Qyz \ v = (q1_ \\ & v).[u]. \end{aligned} $	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge \ Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & QyzE : \text{morph_of } Q \ y \ z \ Qyz \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & v : Q \ y \end{aligned} $ <hr/> $ \begin{aligned} & Qyz \ v = (q1_ \ v).[u] \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>by rewrite Fadjoin_poly_eq // -Dt -sQof2</p>	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \end{aligned} $ <hr/> <p style="text-align: center;">is_realC y Hidden 1 goal(s)</p>

QyzE sQof2 Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$ \begin{aligned} & \text{have /all_sig2[q_} \\ & \text{coqp Dq] v : \{q \mid v} \\ & \text{!= 0} \rightarrow \text{coprimep} \\ & \text{p q \& q}^\wedge \text{Qxz =} \end{aligned} $	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall q1 : \{poly F algebra.vect_ringType (Q z)\}, q1 \setminus is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly F algebra.vect_lalgType (Q x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & \quad t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall x : Q z, } \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & \quad p : \{\text{poly } Q \ x\} \\ & \quad \text{mon_p : p \setminus is monic} \\ & p0_le0 : (p^\wedge QxR).[0] <= 0 \\ & \quad u : Q \ z \\ & \quad y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & \quad Dp : p1 = p^\wedge Qxz \\ & \quad Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun v : Q y} \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & \quad QyzE : \text{forall v : Q y, } Qyz \ v = (q1_ \ v).[u] \\ & \quad v : Q \ y \\ & \hline & \{q : \{\text{poly } Q \ x\} \mid v \neq 0 \rightarrow \text{coprimep } p \ q \ \& \ q^\wedge Qxz = q1_ \ v\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

$q1_ \ v\}$. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p> <i>have</i> /pQwx[q Dq] : q1_ v \is a polyOver Qx by </p>	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & v : Q \ y \\ & q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \\ & Dq : q1_ \ v = q \wedge Qxz \end{aligned} $ <hr/> <p> {q0 : {\text{poly } Q \ x} \mid v != 0 \rightarrow \text{coprimep } p \ q0 \ \& \ q0 \wedge Qxz = q1_ \ v} </p> <p>Hidden 2 goal(s)</p>

apply : Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
Fadjoin_polyOver.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> exists q = > // nz_v; rewrite -(coprimep_map QxzM) -Dp -Dq -gcdp_eq1 </pre>	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly Falgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly Falgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] v : Q y q : {poly Falgebra.vect_lalgType (Q x)} Dq : q1_ v = q ^ Qxz nz_v : v != 0 </pre> <hr/> <pre> gcdp p1 (q1_ v) % = 1 Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>have</p> <p>/minPoly_irr/orP[]</p> <p>// := dvdp_gcdl p1</p>	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly Falgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly Falgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] v : Q y q : {poly Falgebra.vect_lalgType (Q x)} Dq : q1_ v = q ^ Qxz nz_v : v != 0 gcdp p1 (q1_ v) \is a polyOver Qx Hidden 3 goal(s) </pre>

(q1_ v). Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>by rewrite gcdp_polyOver ?minPolyOver</p>	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly Falgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly Falgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] v : Q y q : {poly Falgebra.vect_lalgType (Q x)} Dq : q1_ v = q ^ Qxz nz_v : v != 0 </pre> <hr/> <p>gcdp p1 (q1_ v) % = minPoly Qx u -> gcdp p1 (q1_ v) % = 1 Hidden 2 goal(s)</p>

?Fadjoin_polyOver. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> rewrite -/p1 {1}/eqp dvp_gcd => /and3P[_ _ /dvp_leq/ = </pre>	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly Falgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly Falgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] v : Q y q : {poly Falgebra.vect_lalgType (Q x)} Dq : q1_ v = q ^ Qxz nz_v : v != 0 </pre> <hr/> <pre> (q1_ v != 0) ==> (size p1 <= size (q1_ v))%N -> gcdp p1 (q1_ v) %0 = 1 Hidden 2 goal(s) </pre>

/implyP]Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> rewrite size_minPoly ltnNge size_poly (contraNneq _ </pre>	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly Falgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly Falgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] v : Q y q : {poly Falgebra.vect_lalgType (Q x)} Dq : q1_ v = q ^ Qxz nz_v : v != 0 q1v0 : q1_ v = 0 ----- v == 0 Hidden 2 goal(s) </pre>

nz_v) // Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow F\text{adjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \end{aligned} $ <hr/> $ \begin{aligned} & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

by rewrite

-(fmorph eq0
[rmorphism of
Qyz]) / = QyzE
q1v0 horner0.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_}p : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow F\text{adjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprime } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^\wedge - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \end{aligned} $ <hr/> $ \begin{aligned} & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $
<p>pose h2 : R := 2% : R^\wedge = 1; have nz2 : 2% : R \neq 0 : > R by rewrite pnatr_eq0.</p>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly F algebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly F algebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R is_realC y Hidden 1 goal(s) </pre>

pose itv <= ab.2].
Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
c : R | ab.1 <= c
<= ab.2].

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_}p : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ \hline & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$\text{pose } wid \ ab : R :=$
 $ab.2 - ab.1; \text{ pose}$
 $mid \ ab := (ab.1 +$
 $ab.2) * h2.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_}p : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ \hline & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$$\begin{aligned}
& \text{pose sub_itv } ab \ cd \\
& := cd.1 <= ab.1 : > \\
& R \setminus ab.2 <= cd.2 \\
& : > R.
\end{aligned}$$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^\wedge - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ \hline & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$$\begin{aligned}
& \text{pose } xup \ q \ ab := [\wedge \\
& q.[ab.1] <= 0, \\
& q.[ab.2] >= 0 \ \& \\
& ab.1 <= ab.2 : > \\
& R].
\end{aligned}$$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad \text{is_realC } y \\ & \quad \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

$\text{pose narrow } q \ ab \ (c$
 $:= \text{mid } ab) := \text{if}$
 $q.[c] >= 0 \text{ then}$
 $(ab.1, c) \text{ else } (c,$
 $ab.2).$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \hline & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R k : nat q : {poly R} ab : R * R cd := find k q ab : R * R xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ 74 k)% : R] Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have findP k q ab

(cd := find k q ab) :

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad k : \text{nat} \\ & \quad q : \{\text{poly } R\} \\ & \quad ab : R * R \\ & \quad cd := \text{find } k \ q \ ab : R * R \\ \hline & xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / (2^{\wedge} \\ & \quad 75 \quad k)\% : R] \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^\wedge - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad k : \text{nat} \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ \hline & [\wedge xup \ q \ (\text{find } k \ q \ (a, b)), \text{sub_itv } (\text{find } k \ q \ (a, b)) \ (a, b) \ \& \ wid \\ & \quad (\text{find } k \ q \ (a, b)) = wid \ (a, b) / (2^\wedge k) \% : R] \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_}p : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprime } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \end{aligned} $ <hr/> $ \begin{aligned} & [\wedge xup \ q \ (\text{find } k \ q \ (a, b)), \text{sub_itv } (\text{find } k \ q \ (a, b)) \ (a, b) \ \& \ \text{wid} \\ & (\text{find } k \ q \ (a, b)) = \text{wid}(a, b) / (2^{\wedge} k) \% : R] \rightarrow [\wedge xup \ q \ (\text{narrow } q \\ & (\text{find } k \ q \ (a, b))), \text{sub_itv } (\text{narrow } q \ (\text{find } k \ q \ (a, b))) \ (a, b) \ \& \ \text{wid} \\ & (\text{narrow } q \ (\text{find } k \ q \ (a, b))) = \text{wid}(a, b) / (2^{\wedge} k + 1) \% : R] \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => F\text{adjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 78qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & Ded : wid \ (c, d) = wid \ (a, b) / (2^{\wedge} k) \% : R \end{aligned} $
	$ [\wedge xup \ q \ (narrow \ q \ (c, d)), sub_itv \ (narrow \ q \ (c, d)) \ (a, b) \ \& \ wid $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => F\text{adjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 79qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & Dcd : wid \ (c, d) = wid \ (a, b) / (2^{\wedge} k) \% : R \\ & \quad e := (c + d) / 2\% : R : R \\ & \quad le_ce : c <= e \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 80qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & \quad e := (c + d) / 2\% : R : R \\ & \quad le_ce : c <= e \\ & \quad le_ed : e <= d \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 81qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & \quad e := (c + d) / 2\% : R : R \\ & \quad le_ce : c <= e \\ & \quad le_ed : e <= d \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 82qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & \quad e := (c + d) / 2\% : R : R \\ & \quad le_ce : c <= e \\ & \quad le_ed : e <= d \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \%AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 83qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & \quad e := (c + d) / 2\% : R : R \\ & \quad le_ce : c <= e \\ & \quad le_ed : e <= d \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \quad q : \{\text{poly } R\} \\ & \quad a, b : R \\ & \quad xq_ab : xup \ q \ (a, b) \\ & \quad k : \text{nat} \\ & \quad c, d : R \\ & \quad qc_le0 : q.[c] <= 0 \\ & \quad 84qd_ge0 : 0 <= q.[d] \\ & \quad le_cd : c <= d \\ & \quad le_ac : a <= c \\ & \quad le_db : d <= b \\ & \quad e := (c + d) / 2\% : R : R \\ & \quad le_ce : c <= e \\ & \quad le_ed : e <= d \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \ \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ \hline & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] r : poly_ringType (Num.NumDomain.porder_ringType R) q : {poly Num.NumDomain.porder_ringType R} ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xup q ab -> {n : nat forall x0 : R, x0 \in itv (find n q ab) -> (r * q).[x0] < h2} Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] r : poly_ringType (Num.NumDomain.porder_ringType R) q : {poly Num.NumDomain.porder_ringType R} ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xup q ab -> {n : nat forall x0 : R, x0 \in itv (find n q ab) -> (r * q).[x0] < h2} Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab : xup \ q \ ab \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{` } p.[x] \text{` } <= \text{ub}\} \\ & \text{---} \\ & \{n : \text{nat} \mid \text{forall } x0 : R, x0 \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \text{` } (r * q).[x0] < \\ & \quad h2\} \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab : xup \ q \ ab \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad Mu : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab : xup \ q \ ab \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{` } p.[x] <= \text{ub}\} \\ & \quad Mu : R \\ & MuP : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{` } r.[x] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab : xup \ q \ ab \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad Mu : R \\ & MuP : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab : xup \ q \ ab \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{` } p.[x] <= ub\} \\ & \quad Mu : R \\ & MuP : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{` } r.[x] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab : xup \ q \ ab \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{! } p.[x] <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{! } r.[x] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{! } p.[x] <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{! } r.[x] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{ub : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad Mu : R \\ & MuP : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \\ & MqP : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`p.[x]} <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`r.[x]} <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{!}[p.[x]] <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{!}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`p.[x]} <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`r.[x]} <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{ub : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{!}[p.[x]] <= ub\} \\ & \quad Mu : R \\ & MuP : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{!}[r.[x]] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \\ & MqP : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{! } [p.[x]] <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{! } [r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`p.[x]} <= \text{ub}\} \\ & \quad \text{Mu} : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`r.[x]} <= \text{Mu} \\ & \quad \text{Mq} : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{ub_ab} : \text{forall } p : \{\text{poly } R\}, \ \{\text{ub} : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad Mu : R \\ & \text{MuP} : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \\ & \text{MqP} : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & r : \text{poly_ringType } (\text{Num.NumDomain.porder_ringType } R) \\ & q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\} \\ & ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & ub_ab : \text{forall } p : \{\text{poly } R\}, \ \{ub : R \mid \text{forall } x : R, ab.1 <= x <= \\ & \quad ab.2 \rightarrow \text{`}[p.[x]] <= ub\} \\ & \quad Mu : R \\ & MuP : \text{forall } x : R, ab.1 <= x <= ab.2 \rightarrow \text{`}[r.[x]] <= Mu \\ & \quad Mq : \text{nat} \rightarrow R \\ & MqP : \text{forall } (x : \text{nat}) \ (x0 : R), ab.1 <= x0 <= ab.2 \rightarrow \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \ (r * q).[x]\ < h2\} \\ & \text{is_realC } y \\ & \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \ (r * q).[x]\ < h2\} \\ & \{ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \mid xup \ (p \wedge QxR) \ ab\} \\ & \text{Hidden 2 goal(s)} \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \ (r * q).[x]\ < h2\} \\ & \quad b : R \\ & \text{pb_gt0} : \text{forall } x0 : R, b <= x0 \rightarrow 0 < (p \wedge QxR).[x0] \end{aligned} $
	$ \{ab : \text{Num.NumDomain.porder_ringType } R * $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \ \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \ \text{ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ \text{ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R Iab_ := fun n : nat => itv (ab_ n) : nat -> simpl_pred R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R Iab_ := fun n : nat => itv (ab_ n) : nat -> simpl_pred R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \ \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \\ & Iab_ := \text{fun } n : \text{nat} => itv \ (ab_ \ n) : \text{nat} \rightarrow \text{simpl_pred } R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R Iab_ := fun n : nat => itv (ab_ n) : nat -> simpl_pred R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) => [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) (q : \{\text{poly } R\}) (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \\ & Iab_ := \text{fun } n : \text{nat} => itv \ (ab_ \ n) : \text{nat} \rightarrow \text{simpl_pred } R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \ \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \\ & Iab_ := \text{fun } n : \text{nat} => itv \ (ab_ \ n) : \text{nat} \rightarrow \text{simpl_pred } R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \ \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \\ & Iab_ := \text{fun } n : \text{nat} => itv \ (ab_ \ n) : \text{nat} \rightarrow \text{simpl_pred } R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R Iab_ := fun n : nat => itv (ab_ n) : nat -> simpl_pred R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \\ & Iab_ := \text{fun } n : \text{nat} => itv \ (ab_ \ n) : \text{nat} \rightarrow \text{simpl_pred } R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R Iab_ := fun n : nat => itv (ab_ n) : nat -> simpl_pred R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \\ & Iab_ := \text{fun } n : \text{nat} => itv \ (ab_ \ n) : \text{nat} \rightarrow \text{simpl_pred } R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^\wedge - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^\wedge k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ x \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & \quad 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & \quad q \ ab \text{ in } xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \ \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in itv } (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2\% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2\% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad mid := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & sub_itv := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & narrow := \text{fun } (q : \{\text{poly } R\}) (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & find := \text{fun } (k : \text{nat}) (q : \{\text{poly } R\}) => \text{iter } k \ (narrow \ q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & findP : \text{forall } (k : \text{nat}) (q : \{\text{poly } R\}) (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, sub_itv \ cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k)\% : R] \\ & find_root : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \quad (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \quad (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle \langle 1; \text{in } Q \ z \ x \rangle \rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle \langle 1; t_ \ x \rangle \rangle = \langle \langle Qx; x \rangle \rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly } \text{Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in } itv \ (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y \Rightarrow \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R \Rightarrow [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R \Rightarrow ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R \Rightarrow (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R \Rightarrow (\text{cd}.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (\text{cd}.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \Rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) \Rightarrow \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) \Rightarrow \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in itv } (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \text{ab0} : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \text{xab0} : xup \ (p \wedge QxR) \text{ ab0} \\ & \text{ab_} := \text{fun } n : \text{nat} \Rightarrow \text{find } n \ (p \wedge QxR) \text{ ab0} : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	$ \begin{aligned} & x : C \\ & R : \text{archiFieldType} \\ & QxR : \{\text{rmorphism } Q \ x \rightarrow R\} \\ & z : C \\ & Qxz : \text{!AHom}(Q \ x, Q \ z) \\ & QxzE : \text{morph_of } Q \ z \ Qxz \\ & Dx : \text{of } Q \ z \ (\text{in } Q \ z \ x) = x \\ & Qx := \langle\langle 1; \text{in } Q \ z \ x \rangle\rangle \% AS : \{\text{subfield } Q \ z\} \\ & QxzM := [\text{rmorphism of } Qxz] : \{\text{rmorphism } Q \ x \rightarrow Q \ z\} \\ & pQwx : \text{forall } q1 : \{\text{poly } F\text{algebra.vect_ringType } (Q \ z)\}, \ q1 \setminus \text{is a} \\ & \text{polyOver } Qx \rightarrow \{q : \{\text{poly } F\text{algebra.vect_lalgType } (Q \ x)\} \mid q1 = q \\ & \quad \wedge Qxz\} \\ & t_ : Q \ z \rightarrow Q \ z \\ & Dt : \text{forall } x : Q \ z, \langle\langle 1; t_ \ x \rangle\rangle = \langle\langle Qx; x \rangle\rangle \\ & p : \{\text{poly } Q \ x\} \\ & \text{mon_p} : p \setminus \text{is monic} \\ & p0_le0 : (p \wedge QxR).[0] <= 0 \\ & u : Q \ z \\ & y := \text{of } Q \ z \ (t_ \ u) : C \\ & p1 := \text{minPoly } Qx \ u : \{\text{poly } Q \ z\} \\ & Dp : p1 = p \wedge Qxz \\ & Qyz : \text{!AHom}(Q \ y, Q \ z) \\ & q1_ := \text{fun } v : Q \ y => \text{Fadjoin_poly } Qx \ u \ (Qyz \ v) : Q \ y \rightarrow \{\text{poly} \\ & \quad Q \ z\} \\ & QyzE : \text{forall } v : Q \ y, Qyz \ v = (q1_ \ v).[u] \\ & q_ : Q \ y \rightarrow \{\text{poly } Q \ x\} \\ & coqp : \text{forall } x0 : Q \ y, x0 \neq 0 \rightarrow \text{coprimep } p \ (q_ \ x0) \\ & Dq : \text{forall } x0 : Q \ y, q_ \ x0 \wedge Qxz = q1_ \ x0 \\ & h2 := (2 \% : R^{\wedge} - 1 : R) : R \\ & nz2 : 2 \% : R \neq 0 \\ & itv := \text{fun } ab : R * R => [\text{pred } c \mid ab.1 <= c \ \& \ c <= ab.2] : R * R \\ & \quad \rightarrow \text{simpl_pred } R \\ & wid := ((\text{fun } ab : R * R => ab.2 - ab.1) : R * R \rightarrow R) : R * R \rightarrow R \\ & \quad \text{mid} := \text{fun } ab : R * R => (ab.1 + ab.2) * h2 : R * R \rightarrow R \\ & \text{sub_itv} := \text{fun } ab \ cd : R * R => (cd.1 : R) <= (ab.1 : R) \wedge (ab.2 : \\ & \quad R) <= (cd.2 : R) : R * R \rightarrow R * R \rightarrow \text{Prop} \\ & xup := \text{fun } (q : \{\text{poly Num.NumDomain.porder_ringType } R\}) \\ & \quad (ab : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R) \rightarrow [\wedge q.[ab.1] <= 0, 0 \\ & \quad <= q.[ab.2] \ \& \ (ab.1 : R) <= (ab.2 : R)] : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\} \\ & \quad \rightarrow \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \rightarrow \text{Prop} \\ & \text{narrow} := \text{fun } (q : \{\text{poly } R\}) \ (ab : R * R) => \text{let } c := \text{mid } ab \text{ in if} \\ & 0 <= q.[c] \text{ then } (ab.1, c) \text{ else } (c, ab.2) : \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{find} := \text{fun } (k : \text{nat}) \ (q : \{\text{poly } R\}) => \text{iter } k \ (\text{narrow } q) : \text{nat} \rightarrow \\ & \quad \{\text{poly } R\} \rightarrow R * R \rightarrow R * R \\ & \text{findP} : \text{forall } (k : \text{nat}) \ (q : \{\text{poly } R\}) \ (ab : R * R), \text{let } cd := \text{find } k \\ & q \ ab \text{ in } \ xup \ q \ ab \rightarrow [\wedge xup \ q \ cd, \text{sub_itv } cd \ ab \ \& \ wid \ cd = wid \ ab / \\ & \quad (2^{\wedge} k) \% : R] \\ & \text{find_root} : \text{forall } (r : \text{poly_ringType} \\ & \quad (\text{Num.NumDomain.porder_ringType } R)) \ (q : \{\text{poly} \\ & \quad \text{Num.NumDomain.porder_ringType } R\}) \ (ab : \\ & \quad \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R), \ xup \ q \ ab \rightarrow \{n : \text{nat} \mid \\ & \quad \text{forall } x : R, x \setminus \text{in itv } (\text{find } n \ q \ ab) \rightarrow \setminus (r * q).[x] < h2\} \\ & \quad ab0 : \text{Num.NumDomain.porder_ringType } R * \\ & \quad \text{Num.NumDomain.porder_ringType } R \\ & \quad xab0 : xup \ (p \wedge QxR) \ ab0 \\ & ab_ := \text{fun } n : \text{nat} => \text{find } n \ (p \wedge QxR) \ ab0 : \text{nat} \rightarrow R * R \end{aligned} $

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => FAdjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> x : C R : archiFieldType QxR : {rmorphism Q x -> R} z : C Qxz : !AHom(Q x, Q z) QxzE : morph_of Q x z Qxz Dx : of Q z (in Q z x) = x Qx := <<1; in Q z x>>%AS : {subfield Q z} QxzM := [rmorphism of Qxz] : {rmorphism Q x -> Q z} pQwx : forall q1 : {poly FAlgebra.vect_ringType (Q z)}, q1 \is a polyOver Qx -> {q : {poly FAlgebra.vect_lalgType (Q x)} q1 = q ^ Qxz} t_ : Q z -> Q z Dt : forall x : Q z, <<1; t_ x>> = <<Qx; x>> p : {poly Q x} mon_p : p \is monic p0_le0 : (p ^ QxR).[0] <= 0 u : Q z y := of Q z (t_ u) : C p1 := minPoly Qx u : {poly Q z} Dp : p1 = p ^ Qxz Qyz : !AHom(Q y, Q z) q1_ := fun v : Q y => Fadjoin_poly Qx u (Qyz v) : Q y -> {poly Q z} QyzE : forall v : Q y, Qyz v = (q1_ v).[u] q_ : Q y -> {poly Q x} coqp : forall x0 : Q y, x0 != 0 -> coprimep p (q_ x0) Dq : forall x0 : Q y, q_ x0 ^ Qxz = q1_ x0 h2 := (2% : R^ - 1 : R) : R nz2 : 2% : R != 0 itv := fun ab : R * R => [pred c ab.1 <= c & c <= ab.2] : R * R -> simpl_pred R wid := ((fun ab : R * R => ab.2 - ab.1) : R * R -> R) : R * R -> R mid := fun ab : R * R => (ab.1 + ab.2) * h2 : R * R -> R sub_itv := fun ab cd : R * R => (cd.1 : R) <= (ab.1 : R) /\ (ab.2 : R) <= (cd.2 : R) : R * R -> R * R -> Prop xup := fun (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R) => [/\ q.[ab.1] <= 0, 0 <= q.[ab.2] & (ab.1 : R) <= (ab.2 : R)] : {poly Num.NumDomain.porder_ringType R} -> Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R -> Prop narrow := fun (q : {poly R}) (ab : R * R) => let c := mid ab in if 0 <= q.[c] then (ab.1, c) else (c, ab.2) : {poly R} -> R * R -> R * R find := fun (k : nat) (q : {poly R}) => iter k (narrow q) : nat -> {poly R} -> R * R -> R * R findP : forall (k : nat) (q : {poly R}) (ab : R * R), let cd := find k q ab in xup q ab -> [/\ xup q cd, sub_itv cd ab & wid cd = wid ab / (2 ^ k)% : R] find_root : forall (r : poly_ringType (Num.NumDomain.porder_ringType R)) (q : {poly Num.NumDomain.porder_ringType R}) (ab : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R), xup q ab -> {n : nat forall x : R, x \in itv (find n q ab) -> (r * q).[x] < h2} ab0 : Num.NumDomain.porder_ringType R * Num.NumDomain.porder_ringType R xab0 : xup (p ^ QxR) ab0 ab_ := fun n : nat => find n (p ^ QxR) ab0 : nat -> R * R </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$by\ exists$ $(ArchiFieldType$ $Ry\ archiRy);$ $apply : [rmorphism$ $of\ idfun].$ $have\ some_realC :$ $realC.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $\boxed{add_Root} : forall\ (xR : realC)\ (p : \{poly\ C\})\ (c : C),\ \{yR : realC$ $\mid extendsR\ xR\ yR\ \&\ has_Root\ xR\ p\ c \rightarrow root_in\ yR\ p\}$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj\ \&\ \sim\ conj = 1\ id\}$
$suffices\ /all_sig[f$ $QfK] x : \{a \mid$ $in_alg\ (Q\ 0)\ a =$ $x\}.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $f : Q\ 0 \rightarrow rat_Ring$ $QfK : forall\ x : Q\ 0, in_alg\ (Q\ 0)\ (f\ x) = x$ <hr/> $realC$ Hidden 1 goal(s)
$exists\ 0,$ $[archiFieldType\ of$ $rat], f.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $f : Q\ 0 \rightarrow rat_Ring$ $QfK : forall\ x : Q\ 0, in_alg\ (Q\ 0)\ (f\ x) = x$ <hr/> $rmorphism\ f$ Hidden 2 goal(s)
$exact :$ $can2_rmorphism$ $(inj_can_sym$ $QfK\ (fmorph_inj$ $))\ QfK.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $x : Q\ 0$ <hr/> $\{a : rat_Ring \mid in_alg\ (Q\ 0)\ a = x\}$ Hidden 1 goal(s)
$have$ $/Fadjoin1_polyP/sig_eqW[q] :$ $x \setminus in\ <<1;$ $0>>\%VS\ by$ $rewrite\ -sQof2$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $x : Q\ 0$ $q : poly_choiceType\ rat_fieldType$ <hr/> $x = q^{\wedge}[0] \rightarrow \{a : rat_Ring \mid in_alg\ (Q\ 0)\ a = x\}$ Hidden 1 goal(s)
$rmorph0.$ $by\ exists\ q.[0];$ $rewrite$ $-horner_map$ $rmorph0.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $\boxed{some_realC} : realC$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj\ \&\ \sim\ conj = 1\ id\}$

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> pose fix xR n : realC := if n isn't n! + 1 then some_realC else if unpickle (nth 0%N x_ n := tag (CodeSeq.decode n!)) 1) isn't Some (p, c) then xR n! else tag (add_Rroot (xR n!)) p c) n%N -> {subset sQ (x_ m) <= sQ (x_ n)}. </pre>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool [xR] := fix xR (n : nat) : realC := match n with 0 => some_realC n! + 1 => match unpickle (nth 0 (CodeSeq.decode n!) 1) with Some (p, c) => tag (add_Rroot (xR n!) p c) None => xR n! end end : nat -> realC </pre> <hr/> <pre> {conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id} </pre> <hr/> <pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool [x_] := fun n : nat => tag (xR n) : nat -> C </pre> <hr/> <pre> {conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id} </pre> <hr/> <pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool m, n : nat </pre> <hr/> <pre> (m <= n)%N -> {subset sQ (x_ m) <= sQ (x_ n)} Hidden 1 goal(s) </pre>
<pre> move/subnK <-; elim : {n} (n - m)%N => // n IHn x /IHn {IHn} Rx. </pre>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool m, n : nat x : C Rx : x \in sQ (x_ (n + m)%N) </pre> <hr/> <pre> x \in sQ (x_ (n + 1 + m)%N) Hidden 1 goal(s) </pre>
<pre> rewrite addSn /x_ /=; case : (unpickle _) => [[p c]] // =. </pre>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool m, n : nat x : C Rx : x \in sQ (x_ (n + m)%N) p : poly_countType C c : C </pre> <hr/> <pre> x \in sQ (tag (ssrfun.s2val (add_Rroot (xR (n + m)%N) p c))) Hidden 1 goal(s) </pre>
<pre> by case : (add_Rroot _ _ _) => yR /= /(sQtrans _ x) =>. </pre>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool [sRle] : forall m n : nat, (m <= n)%N -> {subset sQ (x_ m) <= sQ (x_ n)} </pre> <hr/> <pre> {conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id} </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\begin{aligned} & \text{have } xR\text{root } n \text{ } p \text{ } c : \\ & \text{has_Rroot } (xR \text{ } n) \\ & p \text{ } c \rightarrow \{m \mid n < = \\ & m \ \& \ \text{root_in } (xR \\ & m) \text{ } p\} \% N. \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \text{ } yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \text{ } (tag \text{ } yR) : \text{realC} \\ & \quad \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n : \text{nat} \\ & \quad p : \{poly \text{ } C\} \\ & \quad c : C \end{aligned}$ <hr/> $\begin{aligned} & \text{has_Rroot } (xR \text{ } n) \text{ } p \text{ } c \rightarrow \{m : \text{nat} \mid (n < = m) \% N \ \& \ \text{root_in } (xR \\ & \quad m) \text{ } p\} \\ & \text{Hidden 1 goal(s)} \end{aligned}$
$\begin{aligned} & \text{case/andAP} \Rightarrow Rp \\ & \text{mon_p } Rc \text{ } Dc; \\ & \text{pose } m := \\ & \text{CodeSeq.code } [:: n; \\ & \text{pickle } (p, c)]. \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \text{ } yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \text{ } (tag \text{ } yR) : \text{realC} \\ & \quad \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n : \text{nat} \\ & \quad p : \{poly \text{ } C\} \\ & \quad c : C \\ & \quad Rp : p \setminus \text{is a polyOver } (sQ \text{ } (tag \text{ } (xR \text{ } n))) \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad Rc : c \setminus \text{in } sQ \text{ } (tag \text{ } (xR \text{ } n)) \\ & \quad Dc : p.[0] == -c^{\wedge} + 2 \\ & \quad m := \text{CodeSeq.code } [:: n; \text{pickle } (p, c)] : \text{nat} \end{aligned}$ <hr/> $\begin{aligned} & \{m0 : \text{nat} \mid (n < = m0) \% N \ \& \ \text{root_in } (xR \text{ } m0) \text{ } p\} \\ & \text{Hidden 1 goal(s)} \end{aligned}$
$\begin{aligned} & \text{have } le_n_m : (n \\ & < = m) \% N \text{ by} \\ & \text{apply/ltnW/(allP} \\ & (\text{CodeSeq.ltn_code} \\ & _)) / \text{mem_head.} \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \text{ } yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \text{ } (tag \text{ } yR) : \text{realC} \\ & \quad \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n : \text{nat} \\ & \quad p : \{poly \text{ } C\} \\ & \quad c : C \\ & \quad Rp : p \setminus \text{is a polyOver } (sQ \text{ } (tag \text{ } (xR \text{ } n))) \\ & \quad \text{mon_p} : p \setminus \text{is monic} \\ & \quad Rc : c \setminus \text{in } sQ \text{ } (tag \text{ } (xR \text{ } n)) \\ & \quad Dc : p.[0] == -c^{\wedge} + 2 \\ & \quad m := \text{CodeSeq.code } [:: n; \text{pickle } (p, c)] : \text{nat} \\ & \quad le_n_m : (n < = m) \% N \end{aligned}$ <hr/> $\begin{aligned} & \{m0 : \text{nat} \mid (n < = m0) \% N \ \& \ \text{root_in } (xR \text{ } m0) \text{ } p\} \\ & \text{Hidden 1 goal(s)} \end{aligned}$

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$exists\ m.\ +1;$ $rewrite\ ?leqW\ /\ x_$ $// =$ $CodeSeq.codeK$ $pickleK.$	$extendsR := fun\ xR\ yR : realC => tag\ xR\ \backslash in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ $p : \{poly\ C\}$ $c : C$ $Rp : p\ \backslash is\ a\ polyOver\ (sQ\ (tag\ (xR\ n)))$ $mon_p : p\ \backslash is\ monic$ $Rc : c\ \backslash in\ sQ\ (tag\ (xR\ n))$ $Dc : p.[0] == -\ c\ ^+2$ $m := CodeSeq.code\ [::\ n;\ pickle\ (p,\ c)] : nat$ $le_n_m : (n <= m)\%N$ <hr/> $root_in\ (ssrfun.s2val\ (add_Rroot\ (xR\ m)\ p\ c))\ p$ Hidden 1 goal(s)
$case : (add_Rroot$ $_ _ _) => yR /\ =$ $_;$ $apply;$ $apply/andAP.$	$extendsR := fun\ xR\ yR : realC => tag\ xR\ \backslash in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ $p : \{poly\ C\}$ $c : C$ $Rp : p\ \backslash is\ a\ polyOver\ (sQ\ (tag\ (xR\ n)))$ $mon_p : p\ \backslash is\ monic$ $Rc : c\ \backslash in\ sQ\ (tag\ (xR\ n))$ $Dc : p.[0] == -\ c\ ^+2$ $m := CodeSeq.code\ [::\ n;\ pickle\ (p,\ c)] : nat$ $le_n_m : (n <= m)\%N$ $yR : realC$ <hr/> $[\wedge\ p\ \backslash is\ a\ polyOver\ (sQ\ (tag\ (xR\ m))),\ p\ \backslash is\ monic,\ c\ \backslash in\ sQ\ (tag$ $(xR\ m))\ \&\ p.[0] == -\ c\ ^+2]$ Hidden 1 goal(s)
$by\ split = > //;$ $first\ apply :$	$extendsR := fun\ xR\ yR : realC => tag\ xR\ \backslash in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $\boxed{xRroot} : forall\ (n : nat)\ (p : \{poly\ C\})\ (c : C),\ has_Rroot\ (xR\ n)$ $p\ c \rightarrow \{m : nat \mid (n <= m)\%N\ \&\ root_in\ (xR\ m)\ p\}$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj\ \&\ \sim\ conj = 1\ id\}$
$polyOverS\ Rp;$ $apply /all_and3[Ri_R$ $Ri_i\ def\ Ri]]\ n\ (x$ $:= x_n) : \{z \mid [\wedge$ $x\ \backslash in\ sQ\ z,\ i\ \backslash in\ sQ$ $z\ \&\ <<<<1;\ inQ\ z$ $x>>;\ inQ\ z\ i>> =$ $fullv]\}$.	$extendsR := fun\ xR\ yR : realC => tag\ xR\ \backslash in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ $x := x_n : C$ <hr/> $\{z : C \mid [\wedge\ x\ \backslash in\ sQ\ z,\ i\ \backslash in\ sQ\ z\ \&\ <<<<1;\ inQ\ z\ x>>;\ inQ\ z\ i>>$ $= fullv]\}$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
—	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad n : nat \\ & \quad x := x_n : C \end{aligned}$ <hr/> $\{z : C \mid [\wedge x \setminus in sQ z, i \setminus in sQ z \ \& \ \lll\lll 1; inQ z x>>; inQ z i>> = fullv]\}$ <p>Hidden 1 goal(s)</p>
$\begin{aligned} & have [z /and3P[z_x \\ & z_i _] Dzi] := \\ & PET [:: x; i]. \end{aligned}$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad n : nat \\ & \quad x := x_n : C \\ & \quad z : C \\ & \quad z_x : mem (sQ z) x \\ & \quad z_i : mem (sQ z) i \\ & \quad Dzi : \lll 1 \ \& \ [seq inQ z i \mid i <- [:: x; i]]>>\%VS = fullv \end{aligned}$ <hr/> $\{z0 : C \mid [\wedge x \setminus in sQ z0, i \setminus in sQ z0 \ \& \ \lll\lll 1; inQ z0 x>>; inQ z0 i>> = fullv]\}$ <p>Hidden 1 goal(s)</p>
$\begin{aligned} & by\ exists\ z; \ rewrite \\ & -adjoin_seq1 \\ & -adjoin_cons. \end{aligned}$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad [z_] : nat \rightarrow C \\ & \quad [Ri_R] : forall x : nat, x_x \setminus in sQ (z_ x) \\ & \quad [Ri_i] : forall x : nat, i \setminus in sQ (z_ x) \\ & \quad [defRi] : forall x : nat, \lll\lll 1; inQ (z_ x) (x_ x)>>; inQ (z_ x) \\ & \quad \quad i>> = fullv \end{aligned}$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$
	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad [i_] := fun n : nat => inQ (z_ n) i : forall n : nat, Q (z_ n) \\ & \quad [R_] := fun n : nat => \lll 1; inQ (z_ n) (x_ n)>>\%AS : forall n \\ & \quad : nat, \\ & \quad \quad \{subfield \\ & \quad \quad Q (z_ n)\} \end{aligned}$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$
$\begin{aligned} & have\ memRi\ inQ \\ & (z_Ri) i; poseB \\ & z_i predT; by inQ (z_ \\ & move = n>>\%AS. \\ & rewrite defRi \\ & memvf. \end{aligned}$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad [memRi] : forall n : nat, \lll R_ n; i_ n>> = i predT \end{aligned}$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$have\ sCle\ m\ n : (m \leq n) \% N \rightarrow \{subset\ sQ\ (z_m) \leq sQ\ (z_n)\}.$	$extendsR := fun\ xR\ yR : realC \Rightarrow tag\ xR \setminus in\ sQ\ (tag\ yR) : realC \rightarrow realC \rightarrow bool$ $m, n : nat$ <hr/> $(m \leq n) \% N \rightarrow \{subset\ sQ\ (z_m) \leq sQ\ (z_n)\}$ Hidden 1 goal(s)
$move/sRle \Rightarrow Rmn_ /sQ_inQ[u <-].$	$extendsR := fun\ xR\ yR : realC \Rightarrow tag\ xR \setminus in\ sQ\ (tag\ yR) : realC \rightarrow realC \rightarrow bool$ $m, n : nat$ $Rmn : \{subset\ sQ\ (x_m) \leq sQ\ (x_n)\}$ $u : Q\ (z_m)$ <hr/> $ofQ\ (z_m)\ u \setminus in\ sQ\ (z_n)$ Hidden 1 goal(s)
$have /Fadjoin_polyP[p /polyOverP\ Rp \rightarrow] := memRi\ m$	$extendsR := fun\ xR\ yR : realC \Rightarrow tag\ xR \setminus in\ sQ\ (tag\ yR) : realC \rightarrow realC \rightarrow bool$ $m, n : nat$ $Rmn : \{subset\ sQ\ (x_m) \leq sQ\ (x_n)\}$ $u : Q\ (z_m)$ $p : \{poly\ Falgebra.vect_ringType\ (Q\ (z_m))\}$ $Rp : forall\ i : nat, p_i \setminus in\ R_m$ <hr/> $ofQ\ (z_m)\ p.[i_m] \setminus in\ sQ\ (z_n)$ Hidden 1 goal(s)
$u.$ $rewrite -horner_map inQ_K ?rpred_horner // =; apply/polyOver_poly =$	$extendsR := fun\ xR\ yR : realC \Rightarrow tag\ xR \setminus in\ sQ\ (tag\ yR) : realC \rightarrow realC \rightarrow bool$ $m, n : nat$ $Rmn : \{subset\ sQ\ (x_m) \leq sQ\ (x_n)\}$ $u : Q\ (z_m)$ $p : \{poly\ Falgebra.vect_ringType\ (Q\ (z_m))\}$ $Rp : forall\ i : nat, p_i \setminus in\ R_m$ $j : nat$ <hr/> $ofQ\ (z_m)\ p_j \setminus in\ sQ\ (z_n)$ Hidden 1 goal(s)
$> j_.$ $by\ apply : sQtrans (Ri_R\ n); rewrite Rmn // -(inQ_K$	$extendsR := fun\ xR\ yR : realC \Rightarrow tag\ xR \setminus in\ sQ\ (tag\ yR) : realC \rightarrow realC \rightarrow bool$ $sCle : forall\ m\ n : nat, (m \leq n) \% N \rightarrow \{subset\ sQ\ (z_m) \leq sQ\ (z_n)\}$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$
$_ (Ri_R\ m)) sQof2.$ $have\ Ri\ n : i \setminus notin\ sQ\ (x_n).$	$extendsR := fun\ xR\ yR : realC \Rightarrow tag\ xR \setminus in\ sQ\ (tag\ yR) : realC \rightarrow realC \rightarrow bool$ $n : nat$ <hr/> $i \setminus notin\ sQ\ (x_n)$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre>rewrite /x_ ; case : (xR n) => x [Rn QxR] / = .</pre>	<pre>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool n : nat x : C Rn : archiFieldType QxR : {rmorphism Q x -> Rn}</pre> <hr/> <pre>i \notin sQ x Hidden 1 goal(s)</pre>
<pre>apply : contraL (@ltr01 Rn) => /sQ_inQ[v Di].</pre>	<pre>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool n : nat x : C Rn : archiFieldType QxR : {rmorphism Q x -> Rn} v : Q x Di : ofQ x v = i</pre> <hr/> <pre>~~ (0 < 1) Hidden 1 goal(s)</pre>
<pre>suffices /eqP <- : - QxR v ^+ 2 == 1 by rewrite oppr_gt0 -leNgt sqr_ge0.</pre>	<pre>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool n : nat x : C Rn : archiFieldType QxR : {rmorphism Q x -> Rn} v : Q x Di : ofQ x v = i</pre> <hr/> <pre>- QxR v ^+ 2 == 1 Hidden 1 goal(s)</pre>
<pre>rewrite -rmorphX -rmorphN fmorph_eq1 -(fmorph_eq1 (ofQ x)) rmorphN eqr_oppLR.</pre>	<pre>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool n : nat x : C Rn : archiFieldType QxR : {rmorphism Q x -> Rn} v : Q x Di : ofQ x v = i</pre> <hr/> <pre>ofQ x (v ^+ 2) == -1 Hidden 1 goal(s)</pre>
<pre>by rewrite rmorphX_Di_Di2</pre>	<pre>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool [R'i] : forall n : nat, i \notin sQ (x_ n)</pre> <hr/> <pre>{conj : {rmorphism C -> C} involutive conj & ~ conj = 1 id}</pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>have szX2_1 : size (tX^2 + 1) = 3.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ <hr/> $\text{forall } t : \text{ringType}, \text{size } (tX^2 + 1) = 3$ Hidden 1 goal(s)
<i>by move => R; rewrite size_addl</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $\boxed{\text{szX2_1}} : \text{forall } t : \text{ringType}, \text{size } (tX^2 + 1) = 3$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
<i>?size_polyXn ?size_poly1. have minp_i n (p_i := minPoly (R_n) (i_n)) : p_i = tX^2 + 1.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) (i_n) : \{\text{poly } Q (z_n)\}$ <hr/> $p_i = tX^2 + 1$ Hidden 1 goal(s)
<i>have p_dv_X2_1 : p_i % tX^2 + 1.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) (i_n) : \{\text{poly } Q (z_n)\}$ <hr/> $p_i \% tX^2 + 1$ Hidden 2 goal(s)
<i>rewrite minPoly_dvdp ?rpredD ?rpredX ?rpred1</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) (i_n) : \{\text{poly } Q (z_n)\}$ <hr/> $\text{root } (tX^2 + 1) (i_n)$ Hidden 2 goal(s)
<i>?polyOverX //. rewrite -(fmorph_root (ofQ _)) inQ_K // rmorphD rmorph1 /= map_polyXn.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) (i_n) : \{\text{poly } Q (z_n)\}$ <hr/> $\text{root } (tX^2 + 1) i$ Hidden 2 goal(s)
<i>by rewrite rootE hornerD hornerXn hornerC Di2 addNr.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) (i_n) : \{\text{poly } Q (z_n)\}$ $p_dv_X2_1 : p_i \% tX^2 + 1$ <hr/> $p_i = tX^2 + 1$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>apply/eqP; rewrite -eqp_moniac ?monic_minPoly</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) \ (i_n) : \{\text{poly } Q \ (z_n)\}$ $p_dv_X2_1 : p_i \% \iota X^2 + 1$ <hr/> $\iota X^2 + 1 \setminus \text{is monic}$ Hidden 2 goal(s)
<i>//; last first.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) \ (i_n) : \{\text{poly } Q \ (z_n)\}$ $p_dv_X2_1 : p_i \% \iota X^2 + 1$ <hr/> $p_i \% = \iota X^2 + 1$ Hidden 1 goal(s)
<i>by rewrite monicE lead_coefE szX2_1 coefD coefXn coefC addr0.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $p_i := \text{minPoly } (R_n) \ (i_n) : \{\text{poly } Q \ (z_n)\}$ $p_dv_X2_1 : p_i \% \iota X^2 + 1$ <hr/> $\sim (size \ p_i \leq 2) \% N$ Hidden 1 goal(s)
<i>rewrite -dvdp_size_eqp // eqn_leq dvdp_leq -?size_poly_eq0 ?szX2_1 // = ltnNge.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $\boxed{\text{minp_i}} : \text{forall } n : \text{nat}, \text{ let } p_i := \text{minPoly } (R_n) \ (i_n) \text{ in } p_i = \iota X^2 + 1$ <hr/> $\{\text{conj} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj} \ \& \ \sim \text{conj} = 1 \text{ id}\}$
<i>adjoin_deg_eq1 -sQof2 !inQ_K.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $z : C$ <hr/> $\{n : \text{nat} \mid z \setminus \text{in } sQ \ (z_n)\}$ Hidden 1 goal(s)
<i>have /all_sig[n_ FTA] z : {n z \in sQ (z_n)}.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $z : C$ <hr/> $(\text{forall } z0 : C, i \setminus \text{in } sQ \ z0 \wedge \text{is_Gal } z0 \rightarrow \{n : \text{nat} \mid z0 \setminus \text{in } sQ \ (z_n)\}) \rightarrow \{n : \text{nat} \mid z \setminus \text{in } sQ \ (z_n)\}$ Hidden 2 goal(s)
<i>without loss [z_i gal_z] : z / i \in sQ z /\ is_Gal z.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $z : C$ <hr/> $(\text{forall } z0 : C, i \setminus \text{in } sQ \ z0 \wedge \text{is_Gal } z0 \rightarrow \{n : \text{nat} \mid z0 \setminus \text{in } sQ \ (z_n)\}) \rightarrow \{n : \text{nat} \mid z \setminus \text{in } sQ \ (z_n)\}$ Hidden 2 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{have } [y \text{ /and3P } [sQtrans \text{ } y_z \text{ /sQtrans } y_i \text{ }] \text{ }] := PET [:: z; i].$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad \quad z : C \\ & \quad \quad y : C \\ & \quad y_z : forall s : C, y \setminus in sQ s \rightarrow z \setminus in sQ s \\ & \quad y_i : forall s : C, y \setminus in sQ s \rightarrow i \setminus in sQ s \end{aligned}$ <hr/> $\begin{aligned} & (forall z0 : C, i \setminus in sQ z0 \wedge is_Gal z0 \rightarrow \{n : nat \mid z0 \setminus in sQ (z_n)\}) \rightarrow \{n : nat \mid z \setminus in sQ (z_n)\} \\ & \quad \text{Hidden 2 goal(s)} \end{aligned}$
$\text{have } [t \text{ /sQtrans } t_y \text{ gal_t}] := galQ \text{ } y.$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad \quad z : C \\ & \quad \quad y : C \\ & \quad y_z : forall s : C, y \setminus in sQ s \rightarrow z \setminus in sQ s \\ & \quad y_i : forall s : C, y \setminus in sQ s \rightarrow i \setminus in sQ s \\ & \quad \quad t : C \\ & \quad t_y : forall s : C, t \setminus in sQ s \rightarrow y \setminus in sQ s \\ & \quad gal_t : is_Gal t \end{aligned}$ <hr/> $\begin{aligned} & (forall z0 : C, i \setminus in sQ z0 \wedge is_Gal z0 \rightarrow \{n : nat \mid z0 \setminus in sQ (z_n)\}) \rightarrow \{n : nat \mid z \setminus in sQ (z_n)\} \\ & \quad \text{Hidden 2 goal(s)} \end{aligned}$
$\text{by case } (_ t) => [n]; \text{ last exists } n; \text{ rewrite } ?y_z \text{ } ?y_i \text{ } ?t_y.$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad \quad z : C \\ & \quad \quad z_i : i \setminus in sQ z \\ & \quad gal_z : is_Gal z \end{aligned}$ <hr/> $\begin{aligned} & \{n : nat \mid z \setminus in sQ (z_n)\} \\ & \quad \text{Hidden 1 goal(s)} \end{aligned}$
$\text{apply/sig_eqW;} \\ \text{have } n := 0\%N.$	$\begin{aligned} & extendsR := fun xR yR : realC => tag xR \setminus in sQ (tag yR) : realC \\ & \quad \rightarrow realC \rightarrow bool \\ & \quad \quad z : C \\ & \quad \quad z_i : i \setminus in sQ z \\ & \quad gal_z : is_Gal z \\ & \quad \quad n : nat \end{aligned}$ <hr/> $\begin{aligned} & exists x : nat_choiceType, (z \setminus in sQ (z_x)) = true \\ & \quad \text{Hidden 1 goal(s)} \end{aligned}$

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>have [p] : exists p, [&& p \is monic, root p z & p \is a polyOver (sQ (z_ n))].</p>	<p>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z n : nat</p> <hr/> <p>exists p : {poly C}, [&& p \is monic, root p z & p \is a polyOver (sQ (z_ n))] Hidden 2 goal(s)</p>
<p>have [p mon_p pz0] := algC z; exists (p ^ QtoC).</p>	<p>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z n : nat p : {poly [countFieldType of rat]} mon_p : p \is monic pz0 : root (p ^ QtoC) z</p> <hr/> <p>[&& p ^ QtoC \is monic, root (p ^ QtoC) z & p ^ QtoC \is a polyOver (sQ (z_ n))] Hidden 2 goal(s)</p>
<p>by rewrite map_monc mon_p pz0 -(pQof (z_ n)); apply/polyOver_poly.</p>	<p>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z n : nat p : {poly C}</p> <hr/> <p>[&& p \is monic, root p z & p \is a polyOver (sQ (z_ n))] ->exists x : nat_choiceType, (z \in sQ (z_ x)) = true Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{have } [d \text{ lepd}] := \text{ubnP } (\text{size } p);$ $\text{elim} : d = > // d$ $\text{IHd in } p \text{ n lepd} *$ $=> \text{pz0}.$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} => \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \\ \rightarrow \text{realC} \rightarrow \text{bool}$ $z : C$ $z_i : i \setminus \text{in } sQ \ z$ $\text{gal_z} : \text{is_Gal } z$ $d : \text{nat}$ $\text{IHd} : \text{forall } (p : \{\text{poly } C\}) \ (n : \text{nat}), \ (\text{size } p < d) \% N \rightarrow [\&\& p \\ \setminus \text{is monic, root } p \ z \ \& \ p \setminus \text{is a polyOver } (sQ \ (z_n))] \rightarrow \text{exists } x : \\ \text{nat_choiceType}, (z \setminus \text{in } sQ \ (z_x)) = \text{true}$ $p : \{\text{poly } C\}$ $n : \text{nat}$ $\text{lepd} : (\text{size } p < d. + 1) \% N$ $\text{pz0} : [\&\& p \setminus \text{is monic, root } p \ z \ \& \ p \setminus \text{is a polyOver } (sQ \ (z_n))]$ <hr/> $\text{exists } x : \text{nat_choiceType}, (z \setminus \text{in } sQ \ (z_x)) = \text{true}$ <p>Hidden 1 goal(s)</p>
$\text{have } [t \ [t_C \ t_z \\ \text{gal_t}]] : \text{exists } t,$ $[\wedge \ z_n \setminus \text{in } sQ \ t, z \\ \setminus \text{in } sQ \ t \ \& \ \text{is_Gal}$ $t].$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} => \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \\ \rightarrow \text{realC} \rightarrow \text{bool}$ $z : C$ $z_i : i \setminus \text{in } sQ \ z$ $\text{gal_z} : \text{is_Gal } z$ $d : \text{nat}$ $\text{IHd} : \text{forall } (p : \{\text{poly } C\}) \ (n : \text{nat}), \ (\text{size } p < d) \% N \rightarrow [\&\& p \\ \setminus \text{is monic, root } p \ z \ \& \ p \setminus \text{is a polyOver } (sQ \ (z_n))] \rightarrow \text{exists } x : \\ \text{nat_choiceType}, (z \setminus \text{in } sQ \ (z_x)) = \text{true}$ $p : \{\text{poly } C\}$ $n : \text{nat}$ $\text{lepd} : (\text{size } p < d. + 1) \% N$ $\text{pz0} : [\&\& p \setminus \text{is monic, root } p \ z \ \& \ p \setminus \text{is a polyOver } (sQ \ (z_n))]$ <hr/> $\text{exists } t : C, [\wedge \ z_n \setminus \text{in } sQ \ t, z \setminus \text{in } sQ \ t \ \& \ \text{is_Gal } t]$ <p>Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>have [y /and3P[y_C y_z _] := PET [:: z_ n; z].</p>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] y : C y_C : mem (sQ y) (z_n) y_z : mem (sQ y) z </pre> <hr/> <p><<1 & [seq inQ y i i <- [:: z_n; z]]>>%VS = fullv ->exists t : C, [/ \ z_n \in sQ t, z \in sQ t & is_Gal t] Hidden 2 goal(s)</p>
<p>by have [t /(sQtrans y)t_y] := galQ y; exists t; rewrite !t_y.</p>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>pose Qt := SplittingFieldType rat (Q t) gal_t; have /QtoQ[CnQt CnQtE] := t_C.</p>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>
<p>pose Rn : {subfield Qt} := (CnQt @ : R_n)%AS; pose i_t : Qt := CnQt (i_n).</p>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} </pre> <hr/> <pre> pose Cn : {subfield Qt} := <<Rn; i_t>>%AS </pre>
	<pre> exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>have defCn : Cn = limg CnQt : > {vspace Q t} by rewrite / =</p>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
def Ri.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p><i>have memRn u : (u</i> <i>\in Rn) = (ofQ t u</i> <i>\in sQ (x _ n)).</i></p>	<pre> <i>extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC</i> <i>-> realC -> bool</i> <i>z : C</i> <i>z_i : i \in sQ z</i> <i>gal_z : is_Gal z</i> <i>d : nat</i> <i>IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p</i> <i>\is monic, root p z & p \is a polyOver (sQ (z _ n))]] -> exists x :</i> <i>nat_choiceType, (z \in sQ (z _ x)) = true</i> <i>p : {poly C}</i> <i>n : nat</i> <i>lepd : (size p < d. + 1)%N</i> <i>pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z _ n))]</i> <i>t : C</i> <i>t_C : z _ n \in sQ t</i> <i>t_z : z \in sQ t</i> <i>gal_t : is_Gal t</i> <i>Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType</i> <i>rat_fieldType</i> <i>CnQt : AHom(Q (z _ n), Q t)</i> <i>CnQtE : morph_ofQ (z _ n) t CnQt</i> <i>Rn := ((CnQt @ : R _ n)%AS : {subfield Qt}) : {subfield Qt}</i> <i>i_t := (CnQt (i _ n) : Qt) : Qt</i> <i>Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt}</i> <i>defCn : Cn = limg CnQt</i> <i>u : Qt</i> </pre> <hr/> <p><i>(u \in Rn) = (ofQ t u \in sQ (x _ n))</i> Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$ \begin{aligned} & \text{by rewrite / =} \\ & \text{aimg_adjoin aimg1} \end{aligned} $	$ \begin{aligned} & \text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \\ & \quad \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad \quad z : C \\ & \quad \quad z_i : i \setminus \text{in } sQ \ z \\ & \quad \quad gal_z : \text{is_Gal } z \\ & \quad \quad d : \text{nat} \\ & \text{IHd} : \text{forall } (p : \{\text{poly } C\}) (n : \text{nat}), (size \ p < d) \% N \rightarrow [\&\& \ p \\ & \quad \setminus \text{is monic, root } p \ z \ \& \ p \setminus \text{is a polyOver } (sQ \ (z_n))] \rightarrow \text{exists } x : \\ & \quad \text{nat_choiceType, } (z \setminus \text{in } sQ \ (z_x)) = \text{true} \\ & \quad \quad p : \{\text{poly } C\} \\ & \quad \quad n : \text{nat} \\ & \quad \quad lepd : (size \ p < d. + 1) \% N \\ & \text{pz0} : [\&\& \ p \setminus \text{is monic, root } p \ z \ \& \ p \setminus \text{is a polyOver } (sQ \ (z_n))] \\ & \quad \quad t : C \\ & \quad \quad t_C : z_n \setminus \text{in } sQ \ t \\ & \quad \quad t_z : z \setminus \text{in } sQ \ t \\ & \quad \quad gal_t : \text{is_Gal } t \\ & \text{Qt} := \text{SplittingFieldType rat } (Q \ t) \ gal_t : \text{splittingFieldType} \\ & \quad \text{rat_fieldType} \\ & \quad \text{CnQt} : \text{AHom}(Q \ (z_n), Q \ t) \\ & \quad \text{CnQtE} : \text{morph_of } Q \ (z_n) \ t \ \text{CnQt} \\ & \text{Rn} := ((\text{CnQt} \ @ : R_n) \% AS : \{\text{subfield } Qt\}) : \{\text{subfield } Qt\} \\ & \quad \quad i_t := (\text{CnQt} \ (i_n) : Qt) : Qt \\ & \quad \text{Cn} := (<< Rn; i_t >> \% AS : \{\text{subfield } Qt\}) : \{\text{subfield } Qt\} \\ & \quad \quad \text{defCn} : Cn = \text{limg } CnQt \\ & \text{memRn} : \text{forall } u : Qt, (u \setminus \text{in } Rn) = (\text{of } Q \ t \ u \setminus \text{in } sQ \ (x_n)) \\ & \quad \quad \text{exists } x : \text{nat_choiceType, } (z \setminus \text{in } sQ \ (z_x)) = \text{true} \\ & \quad \quad \text{Hidden 1 goal(s)} \end{aligned} $

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
inQ_K.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) u : Qt </pre> <hr/> <pre> (u \in Cn) = (ofQ t u \in sQ (z_n)) Hidden 2 goal(s) </pre>

have memCn u : (u \in Cn) = (ofQ t u \in sQ (x_n))
 Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) u : Qt v : Q (z_n) Dv : ofQ (z_n) v = z_n genCn : <<1; v>> = fullv </pre> <hr/> <pre> (u \in Cn) = (ofQ t u \in sQ (z_n)) Hidden 2 goal(s) </pre>
<pre> have [v Dv genCn] := genQz (z_n). </pre>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s) </pre>
<pre> by rewrite -Dv =CnQtE sQof2 defCn -genCn aimg_adjoin aimgl. </pre>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i </pre> <hr/> <p style="text-align: center;"><i>exists x : nat_choiceType, (z \in sQ (z_x)) = true</i></p>
<p><i>have Dit : ofQ t</i> <i>i_t = i by require</i> <i>CnQtE inQ_K.</i></p>	<p style="text-align: center;">Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i </pre> <hr/> <p style="text-align: center;"> $i_t^{\wedge} + 2 = -1$ Hidden 2 goal(s) </p>

have Dit2 : $i_t^{\wedge} +$

$2 = -1$. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 </pre> <hr/> <p style="text-align: center;"><i>exists x : nat_choiceType, (z \in sQ (z_x)) = true</i></p>
<p>by apply : (fmorph_inj (ofQ t)); rewrite rmorphX rmorphN1 Dit.</p>	<p style="text-align: center;">Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 \dim_Rn Cn = 2 Hidden 2 goal(s) </pre>

have $\dim_{Rn} Cn = 2$. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>rewrite</i>	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 adjoin_degree (R_n) (i_n) = 2 Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-adjoin_degree_aimg.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 </pre> <hr/> <p style="text-align: center;"><i>exists x : nat_choiceType, (z \in sQ (z_x)) = true</i></p>

by apply :

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 succn_inCn, write
 -size_minPoly
 minp_i.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} </pre> <hr/> <p><i>exists x : nat_choiceType, (z \in sQ (z_x)) = true</i> Hidden 1 goal(s)</p>

have /sQ_inQ[u_z

Dz] := t_z; pose

Rz := <<Cn;

u_z>>%AS.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} </pre> <hr/> <p style="text-align: center;">(\dim_Cn Rz < d)%N Hidden 2 goal(s)</p>

have{p lepd pz0}

te_Rz_d: Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
(\dim_Cn Rz < d)%N.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N pz0 : [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} </pre> <hr/> <pre> (size (minPoly Cn u_z) <= size p)%N Hidden 2 goal(s) </pre>

rewrite -!tnS
-adjoin_degreeE

-size_nCnRn
(leq_trans _ lepd)
// !tnS.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} mon_p : p \is monic pz0 : root p z Cp : p \is a polyOver (sQ (z_n)) </pre> <hr/> <pre> (size (minPoly Cn u_z) <= size p)%N Hidden 2 goal(s) </pre>
<pre> have{pz0} [mon_p pz0 Cp] := and3P pz0. </pre>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} mon_p : p \is monic pz0 : root p z Cp : p \is a polyOver (sQ (z_n)) </pre> <hr/> <pre> have{Cp} Dp : ((p ^ inQ (z_n)) ^ CnQt) ^ ofQ t = p </pre>
	<p>Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} mon_p : p \is monic pz0 : root p z Dp : ((p ^ inQ (z_n)) ^ CnQt) ^ ofQ t = p (size (minPoly Cn u_z) <= size p)%N Hidden 2 goal(s) </pre>
<p>by rewrite -map_poly_comp (eq_map_poly CnQtE) in QpK.</p>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> rewrite -Dp size_map_poly dvdp_leq </pre>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} mon_p : p \is monic pz0 : root p z Dp : ((p ^ inQ (z_n)) ^ CnQt) ^ ofQ t = p </pre> <hr/> <pre> minPoly Cn u_z % (p ^ inQ (z_n)) ^ CnQt Hidden 2 goal(s) </pre>

?monic_?Cn Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-?(map_monie
(ofQ _)) ?Dp //.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true p : {poly C} n : nat lepd : (size p < d. + 1)%N t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} mon_p : p \is monic pz0 : root p z Dp : ((p ^ inQ (z_n)) ^ CnQt) ^ ofQ t = p </pre> <hr/> <pre> rewrite defCn minPoly_dvd p //; </pre>
<pre> try by rewrite -(fmorph_root (ofQ t)) Dz Dp. </pre>	<pre> (p ^ inQ (z_n)) ^ CnQt \is a polyOver (CnQt @ : { : Q (z_n)})%AS Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N </pre> <hr/> <p style="text-align: right;"><i>exists x : nat_choiceType, (z \in sQ (z_x)) = true</i></p>
by apply/polyOver_poly=	Hidden 1 goal(s)

> j _; continue proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
memv_img
?memvf.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))]] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS </p> <hr/> <p> <i>have</i> [sRCn sCnRz] : (Rn <= Cn)%VS /\ (Cn <= Rz)%VS by rewrite !subv_adjoin. </p>	<p> exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s) </p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS </pre> <hr/> <pre> have sRnRz := subv_trans sRCn </pre>
	<pre> exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s) </pre>

sCnRz. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS </pre> <hr/> <p><i>have</i>{gal_z} <i>galRz</i> : <i>galois Rn Rz</i></p>

Rz. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS </pre> <hr/> <pre> exists2 p : {poly Falgebra.vect_ringType Qt}, p \is a polyOver Rn & splittingFieldFor Rn p Rz Hidden 2 goal(s) </pre>

apply/and3P;

split = > //;

apply/splitting_normalField =

> //. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z </pre> <hr/> <pre> pose u : SplittingFieldType rat (Q z) gal_z := </pre>
<i>inQ z z</i> . Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page	<pre> exists2 p : {poly F algebra.vect_ringType Qt}, p \is a polyOver Rn & splittingFieldFor Rn p Rz Hidden 2 goal(s) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt </pre> <hr/> <pre> have /QtoQ[Qzt QztE] := t_z; exists (minPoly 1 u ^ Qzt). </pre>
	<p style="text-align: center;"><i>minPoly 1 u ^ Qzt \is a polyOver Rn</i></p> <p style="text-align: center;">Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<pre> have /polyOver1P[q ->] := minPolyOver 1 u; apply/polyOver_poly= > j _.</pre>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt q : {poly rat_fieldType} j : nat Qzt q ^ @`_j \in Rn Hidden 3 goal(s)</pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt </pre> <hr/> <p style="text-align: center;"><i>splittingFieldFor Rn (minPoly 1 u ^ Qzt) Rz</i> Hidden 2 goal(s)</p>

by rewrite
coef_map

linearZZ rmorph1

rpredZ ?rpred1.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : \AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : \AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (\X - y% : P) </pre> <hr/> <pre> splittingFieldFor Rn (minPoly 1 u ^ Qzt) Rz Hidden 2 goal(s) </pre>

have [s /eqP Ds] :=

splitting_Field_for_Rn (minPoly 1 u ^ Qzt) Rz

1 u.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (!X - y% : P) </pre> <hr/> <p style="text-align: center;"><<Rn & [seq Qzt i i <- s]>>%VS = Rz</p>

rewrite Ds; exists

(map Qzt s); first
by rewrite map_rp
eqppx.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (!X - y% : P) </pre> <hr/> <p style="text-align: center;">(<<Rn & [seq Qzt i i <- s]>> <= Rz)%VS Hidden 3 goal(s)</p>

apply/eqP; rewrite

eqEsubv; Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

apply/andP; split.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z <i>Rz</i> := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (tX - y% : P) w : Q z s_w : w \in s </pre>
<i>apply/Fadjoin_seqP;</i>	<pre> Qzt w \in Rz Hidden 3 goal(s) </pre>

split => //
/mapP[w s_w ->]. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : \AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : \AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (\X - y% : P) (Rz <= <<Rn & [seq Qzt i i <- s]>>)%VS Hidden 2 goal(s) </pre>

by rewrite (subvP

(adjoinS (u_z
(sublv _))) //
-sQof2 Dz QztE.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (!X - y% : P) </pre> <hr/> <pre> i_t \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; u_z>>%AS Hidden 3 goal(s) </pre>

rewrite / =

adjoinC Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
(*Fadjoin_idP* _)
-/Rz; last first.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (!X - y% : P) </pre> <hr/> <pre> (<<CnQt @ : <<1; inQ (z_n) (x_n)>>; u_z>>%AS <= <<CnQt @ : <<1; inQ (z_n) (x_n)>> & [seq Qzt i i <- s]>>)%VS Hidden 2 goal(s) </pre>

by rewrite (subvP
(adjoinSl (sublv
_))) // -sQ of 2 Dz
Dit.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : !AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (!X - y% : P) </pre> <hr/> <pre> u_z \in [seq Qzt i i <- s] Hidden 2 goal(s) </pre>

rewrite / =
-adjoin_seq1

adjoin_seqSr. //:
Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
apply/allP = >
/ =; rewrite
andbT.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z gal_z : is_Gal z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : \AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS u := (inQ z z : SplittingFieldType rat (Q z) gal_z) : SplittingFieldType rat (Q z) gal_z Qzt : \AHom(Q z, Q t) QztE : morph_ofQ z t Qzt s : seq (SplittingFieldType rat (Q z) gal_z) Ds : minPoly 1%AS u = \prod_ (y <- s) (\X - y% : P) </pre> <hr/> <pre> exists2 x : Q z, x \in s & ofQ t u_z = ofQ z x Hidden 2 goal(s) </pre>

rewrite

-(mem_map

(fmorph_inj; ofQ

_))) -map_comp

(eq_map QztE);

apply/mapP.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>by exists u; rewrite ?inQ_K // -root_prod_XsubC</p>	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

-Ds Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
root_minPoly.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz </pre>
<p>have galCz : galois Cn Rz by rewrite (galoisS_galRz)</p>	<hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

?sRCn. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<p>have [Cz C'z] := boolP (u_z \in Cn); first by</p>	<pre> extendsR := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz C'z : u_z \notin Cn </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

exists n; Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-Dz -memCn.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz Ctz : u_z \notin Cn G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) G != 1%g Hidden 2 goal(s) </pre>

pose G := !Gal(Rz /

Cn)%G; Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
ntG : G != 1%g.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz Ctz : u_z \notin Cn G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) </pre> <hr/> <pre> \dim_ <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> <<<<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>>; u_z>> != 1 Hidden 2 goal(s) </pre>
<pre> rewrite triv_card1 -galois_dim </pre>	

1?(galoisContinuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
galCz) ?subvv
// = .

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g </pre> <hr/> <p>exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 1 goal(s)</p>

by rewrite

~~adjoin_degreeE~~

~~adjoin_deg_eq~~

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop </pre> <hr/> <p style="text-align: center;"><i>exists x : nat_choiceType, (z \in sQ (z_x)) = true</i></p>

pose extRz m :=

exists2 w, ofQ t.w
\in sQ (z_m) & w
\in [predD Rz &
Cn].

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~~ mem Cn w Rz_w : mem Rz w </pre> <hr/> <p><i>suffices</i> [m le_n_m [w Cw exists x : nat_choiceType, (z \in sQ (z_x)) = true Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 $/andP[Cw, Rz_w]] : exists2$
 $m, (n <= m)\%N \&$
 $extRz\ m.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} (size (p ^ ofQ t)%R < d)%N Hidden 3 goal(s) </pre>

pose p := minPoly

<<Cn; w>> Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

apply : (*IHd* (p ^
ofQ t) m).

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Cw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} (adjoin_degree <<Cn; w>> u_z < \dim_Cn Rz)%N Hidden 3 goal(s) </pre>

apply : leq_trans

le_Rz_Cn : Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
size_map_poly
size_minPoly ltnS.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Cw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} (\dim_<<Cn; w>>%AS <<Cn; u_z>> < \dim_Cn Rz)%N Hidden 3 goal(s) </pre>

rewrite

adjoin_degreeE

adjoinC Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

(addv_idPl Rz_w)

agenv_id.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Cw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} (\dim <<Cn; u_z>> < \dim <<Cn; w>>%AS * \dim_Cn Rz)%N Hidden 3 goal(s) </pre>

rewrite ltn_divLR

?adim_gt0 //

mulnC.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Cw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} </pre> <hr/> <p>(1 < \dim_Cn <<Cn; w>>%AS)%N Hidden 3 goal(s)</p>

rewrite

muln_divCA

?field_denS

?subv_adjoin //

ltn_Pmulr

?adim_gt0 //.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} </pre> <hr/> <pre> [&& p ^ ofQ t \is monic, root (p ^ ofQ t) z & p ^ ofQ t \is a polyOver (sQ (z_m))] Hidden 2 goal(s) </pre>

by rewrite

~~adjoin_degreeE~~

ltnNge leq_eqVlt Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

orbF

adjoin_deg_eq1.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} p ^ ofQ t \is a polyOver (sQ (z_m)) Hidden 2 goal(s) </pre>

rewrite map_monoid

monic_minPoly

-Dz f morph_root

root_minPoly / = .

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} Cw_p : forall i : nat, p`_i \in <<Cn; w>> p ^ ofQ t \is a polyOver (sQ (z_m)) Hidden 2 goal(s) </pre>

have /polyOverP

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Cw_p : *p* is a polyOver <<Cn;
w>>%VS by
apply :
minPolyOver.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} Cw_p : forall i : nat, p`_i \in <<Cn; w>> q : {poly F algebra.vect_ringType Qt} Cq : q \is a polyOver Cn </pre> <hr/> <pre> ofQ t q.[w] \in sQ (z_m) Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 apply/polyOver_poly =
 > j _; have
 /Fadjoin_polyP[q
 Cq {j} ->] :=
 Cw_p j.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop m : nat le_n_m : (n <= m)%N w : Q t Cw : ofQ t w \in sQ (z_m) Ctw : ~ mem Cn w Rz_w : mem Rz w p := minPoly <<Cn; w>> u_z : {poly Qt} Cw_p : forall i : nat, p`_i \in <<Cn; w>> q : {poly Falgebra.vect_ringType Qt} Cq : q \is a polyOver Cn j : nat </pre> <hr/> <pre> ofQ t q`_j \in sQ (z_m) Hidden 2 goal(s) </pre>

rewrite Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-horner_map
rpred_horner //;
apply/polyOver_poly =
> j _.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop exists2 m : nat, (n <= m)%N & extRz m Hidden 1 goal(s) </pre>

by rewrite (sCle.n)

// -memCn

(polyOverP Cq).

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g extRz := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G ----- exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>

have [evenC|oddC] proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
:= boolP (2. - group
G); last first.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat #(!Gal(Rz / Rn))%G : P ----- exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>

have [P Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
/and3P[sPG evenP
oddPG]] :=
Sylow_exists 2
!Gal(Rz / Rn).

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : <<1; w>> = [aspace of fixedField P] </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
have [w defQw] :=
PET_Qz t [aspace
of fixedField P].

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

pose pw := minPoly
Rn w; pose p := (-
pw * (pw \Po -
!X)) ^ ofQ t.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} </pre> <hr/> <p>(size pw). - 1 = # !Gal(Rz / Rn) : P Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have sz_pw : (size
pw). - 1 =
#|!Gal(Rz / Rn) :
P|.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} \dim_(CnQt @ : <<1; inQ (z_n) (x_n)>>) <<CnQt @ : <<1; inQ (z_n) (x_n)>>; w>> = \dim_(CnQt @ : <<1; inQ (z_n) (x_ n)>>) <<1; w>> Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 305
 rewrite
 size_minPoly
 adjoin_degreeE
 -dim_fixed_galois
 // = -defQw.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} </pre> <hr/> <pre> (<<CnQt @ : <<1; inQ (z_n) (x_n)>>; w>> <= <<1; w>>)%VS Hidden 3 goal(s) </pre>

congr (\dim_Rn
_);
apply/esym/eqP;
rewrite eqEsubv
adjoinSl ?sub1v
// = .

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by apply/FadjoinP;
rewrite
memv_adjoin / =
defQw
-galois_connection.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P </pre> <hr/> <p style="text-align: center;"> <i>p \is monic</i> Hidden 3 goal(s) </p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
308

have mon_p : p \is monic.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_pw : pw \is monic p \is monic Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have mon_pw : pw
 \is monic :=
 monic_minPoly _
 _ .

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_pw : pw \is monic lead_coef (- (pw \Po - !X)) == 1 Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

rewrite
map_monc mulNr
-mulrN monicMl
// monicE.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_pw : pw \is monic - (lead_coef pw * (- lead_coef !X) ^ + (size pw). - 1) == 1 Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

```

rewrite
!(lead_coef_opp,
lead_coef_comp)
?size_opp
?size_polyX //.

```

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by rewrite
lead_coefX sz_pw
-sigr_odd
odd_2mat oddPG
mulrN1 opprK.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic p.[0] = - ofQ t pw.[0] ^ + 2 Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have Dp0 : p.[0] =
- ofQ t pw.[0] ^ +
2.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic ofQ t (- pw).[0] * ofQ t (pw \Po - !X).[0] = - ofQ t pw.[0] ^ + 2 Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

rewrite -(rmorph0
 (ofQ t))
 horner_map
 hornerM
 rmorphM.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m</p> <p style="text-align: center;">315 Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

by rewrite
horner_comp
!hornerN hornerX
oppr0 rmorphN

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 Rpw : pw \is a polyOver Rn </pre> <hr/> <p>exists2 nat, (n <= m)%N & extRz m</p> <p>Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have Rpw : pw \is a
polyOver Rn by

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 Rpw : pw \is a polyOver Rn </pre> <hr/> <p> <i>p</i> is a polyOver (sQ (x_n)) Hidden 3 goal(s) </p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have Rp : p \is a
polyOver (sQ (x

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 Rpw : pw \is a polyOver Rn - pw * (pw \Po - !X) \is a polyOver (CnQt @ : (<<1; inQ (z_n) (x_n)>>%VS Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

apply/polyOver_poly =
> j _; rewrite

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 Rpw : pw \is a polyOver Rn Rp : p \is a polyOver (sQ (x_n)) </pre> <hr/> <p>exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 Rpw : pw \is a polyOver Rn Rp : p \is a polyOver (sQ (x_n)) Rp0320fQ t pw.[0] \in sQ (x_n) </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have Rp0 : ofQ t

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P mon_p : p \is monic Dp0 : p.[0] = - ofQ t pw.[0] ^ + 2 Rpw : pw \is a polyOver Rn Rp : p \is a polyOver (sQ (x_n)) Rp032pfQ t pw.[0] \in sQ (x_n) </pre> <hr/> <pre> has_Rroot (xR n) p (ofQ t pw.[0]) Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N p_Rm_0 : root_in (xR m) p 322 exists2 m0 : nat, (n <= m0)%N & extRz m0 Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N p_Rm_0 : root_in (xR m) p </pre> <hr/> <p style="text-align: center;">323</p> <p style="text-align: center;">{y : C y \in sQ (x_m) & root (pw ^ ofQ t) y}</p> <p style="text-align: center;">Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

have{p_Rm_0} [y
Ry pw_y] : {y | y

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N p_Rm_0 : root_in (xR m) p 324 y : C Ry : y \in sQ (tag (xR m)) </pre> <hr/> <pre> root p y -> exists2 x : C, x \in sQ (x_m) & root (pw ^ ofQ t) x Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental _ Theorem _ of _ Algebras
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N p_Rm_0 : root_in (xR m) p 325 y : C Ry : y \in sQ (tag (xR m)) </pre> <hr/> <pre> root (- pw ^ ofQ t * (pw ^ ofQ t \Po - !X)) y ->exists2 x : C, x \in sQ (x_m) & root (pw ^ ofQ t) x Hidden 3 goal(s) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N p_Rm_0 : root_in (xR m) p 326 y : C Ry : y \in sQ (tag (xR m)) </pre> <hr/> <pre> root (pw ^ ofQ t) y root (pw ^ ofQ t) (- y) ->exists2 x : C, x \in sQ (x_m) & root (pw ^ ofQ t) x Hidden 3 goal(s) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 32Ry : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y exists2 m0 : nat, (n <= m0)%N & extRz m0 Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 328y : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y </pre> <hr/> <p style="text-align: center;"><i>exists2</i> u : Qt, u \in Rz & y = ofQ t u</p> <p style="text-align: center;">Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 32By : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y Rz_w : w \in Rz </pre> <hr/> <p style="text-align: center;"><i>exists2</i> u : Qt, u \in Rz & y = ofQ t u Hidden 3 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 33Ry : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y Rz_w : w \in Rz sg : seq (gal_of Rz) Gsg : sg \subset !Gal(Rz / Rn) Dpw : minPoly Rn w = \prod_ (b <- [seq x w x <- sg]) (!X - b% : P) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 33Ry : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y Rz_w : w \in Rz sg : seq (gal_of Rz) Gsg : sg \subset !Gal(Rz / Rn) s := [seq x w x <- sg] : seq Qt Dpw : minPoly Rn w = \prod_ (b <- s) (!X - b% : P) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 33By : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y Rz_w : w \in Rz sg : seq (gal_of Rz) Gsg : sg \subset !Gal(Rz / Rn) s := [seq x w x <- sg] : seq Qt Dpw : minPoly Rn w = \prod_ (b <- s) (!X - b% : P) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 33Ry : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y Rz_w : w \in Rz sg : seq (gal_of Rz) Gsg : sg \subset !Gal(Rz / Rn) s := [seq x w x <- sg] : seq Qt Dpw : minPoly Rn w = \prod_ (b <- s) (!X - b% : P) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 33Ry : y \in sQ (x_m) pw_y : root (pw ^ ofQ t) y u : Qt Rz_u : u \in Rz Dy : y = ofQ t u </pre> <hr/> <p><i>exists2</i> m0 : nat, (n <= m0)%N & <i>extRz</i> m0</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 33By : y \in sQ (x_m) u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u </pre>
	<pre> exists2 m0 : nat, (n <= m0)%N & extRz m0 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop oddG : ~ 2. - group G P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C 336y : y \in sQ (x_m) u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u </pre>
	<hr/> <p>$u \in [\text{predD } Rz \ \& \ Cn]$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) 337 u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	2. - group G

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) 338 u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	2. - group !Gal(Rz / Rn) -> 2. - group G

Table 1: Proof of Theorem Fundamental _ Theorem _ of _ Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) 339 u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	$(\dim_Cn\ Rz\ \% \ \dim_Rn\ Rz)\ \%N$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) 340 u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	$(\dim Rz \% \dim_Rn Rz * \dim Cn) \% N$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G evenP : 2. - group P oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) 341 u : Qt Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	2. - group !Gal(Rz / Rn)

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^i). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 342 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre> <hr/> <p style="text-align: center;"> <i>P</i> == !Gal(Rz / Rn) Hidden 2 goal(s) </p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 343 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	<p>2. - nat (size pw). - 1 Hidden 2 goal(s)</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 344 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in (<<CnQt @ : (<<1; inQ (z_n) (x_n)>>; i_t>> pu := minPoly Rn u : {poly Qt} </pre>
	<p>(pu % = pw) (pu % = 1)</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 345 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	<pre> (let pu := minPoly Rn u in (pu % = pw) (pu % = 1)) -> 2. - nat (size pw). - 1 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 346 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	<pre> 2. - nat (size (minPoly (CnQt @ : <<1; inQ (z_n) (x_n)>> u)). - 1 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 347 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre>
	<pre> (\dim_(CnQt @ : <<1; inQ (z_n) (x_n)>>)%AS <<(CnQt @ : <<1; inQ (z_n) (x_n)>>)%AS; </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop P : {group gal_finGroupType Rz} sPG : P \subset (!Gal(Rz / Rn))%G oddPG : (2^!). - nat # (!Gal(Rz / Rn))%G : P w : Q t defQw : (<<1; w>> = [aspace of fixedField P] pw := minPoly Rn w : {poly Qt} p := (- pw * (pw \Po - !X)) ^ ofQ t : {poly C} sz_pw : (size pw). - 1 = # !Gal(Rz / Rn) : P Rpw : pw \is a polyOver Rn m : nat lenm : (n <= m)%N y : C Ry : y \in sQ (x_m) u : Qt 348 Rz_u : u \in Rz Dy : y = ofQ t u pw_u : root pw u Cu : u \in <<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>> </pre> <hr/> <pre> (<<(CnQt @ : <<1; inQ (z_n) (x_n)>>)%AS; u>>%AS <= Cn)%VS </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z <i>Rz</i> := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz <i>G</i> := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G ----- exists2 m : nat, (n <= m)%N & extRz m Hidden 1 goal(s) </pre>

exact/FieldCoq. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G </pre> <hr/> <p style="text-align: center;"><i>exists2 w : Qt, w \in Rz & adjoin_degree Cn w = 2</i> Hidden 2 goal(s)</p>

have [w Rz_w
deg_w] : exists2 w,
w \in Rz &
adjoin_degree Cn
w = 2.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebras
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z <i>Rz</i> := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz <i>G</i> := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G </pre>
<pre> have [P sPG iPG] : exists2 P : {group </pre>	<pre> exists2 P : {group gal_of Rz}, P \subset G & # G : P = 2 Hidden 3 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebras on the next page
 $\backslash_{\text{subset } G \text{ \& } \#|G :$
 $|P| = 2.$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G k : nat oG : # G = (2 ^ k. + 1)%N exists2 P : {group gal_of Rz}, P \subset G & # G : P = 2 Hidden 3 goal(s) </pre>

have [____[k oG]]

:= pgroup Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
evenG ntG.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2 - group G k : nat oG : # G = (2 ^ k. + 1)%N P : {group gal_finGroupType Rz} sPG : P \subset G oP : # P = (2 ^ (logn 2 # G). - 1)%N </pre> <hr/> <pre> <i>have</i> [P [sPG _ oP]] := </pre>
<pre> normal_Over evenG (normal_refl G) (leq_pred _). </pre>	<pre> exists2 P0 : {group gal_of Rz}, P0 \subset G & # G : P0 = 2 Hidden 3 goal(s) </pre>

normal_Over continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
evenG
(normal_refl G)
(leq_pred _).

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G P : {group gal_of Rz} sPG : P \subset G iPG : # G : P = 2 </pre>
<pre> by exists P => //; rewrite -divgS // oP oG p_factorK // -expnB ?subSnn. </pre>	<pre> exists2 w : Qt, w \in Rz & adjoin_degree Cn w = 2 Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G P : {group gal_of Rz} sPG : P \subset G iPG : # G : P = 2 w : Q t defQw : <<1; w>> = [aspace of fixedField P] </pre> <hr/> <pre> have [w defQw] := PET_Qz [aspace of fixedField </pre>
Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page	<pre> exists2 w0 : Qt, w0 \in Rz & adjoin_degree Cn w0 = 2 Hidden 2 goal(s) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat IHd : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t Qt := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt Rn := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt memRn : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) memCn : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g extRz := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G P : {group gal_of Rz} sPG : P \subset G iPG : # G : P = 2 w : Q t defQw : <<1; w>> = [aspace of fixedField P] </pre>
<i>exists w; first by rewrite</i>	<pre> adjoin_degree Cn w = 2 Hidden 2 goal(s) </pre>

—sub_adjoinlv
defQw capvSt.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G P : {group gal_of Rz} sPG : P \subset G iPG : # G : P = 2 w : Q t defQw : <<1; w>> = [aspace of fixedField P] </pre>
<pre> rewrite adjoin_degreeE -iPG </pre>	<hr/> <pre> <<Cn; w>> = <<1; w>> Hidden 2 goal(s) </pre>

-dim_fixed_galois
Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
// -defQw; congr
(\dim_Cn _).

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G P : {group gal_of Rz} sPG : P \subset G iPG : # G : P = 2 w : Q t defQw : <<1; w>> = [aspace of fixedField P] </pre> <hr/> <pre> (<<CnQt @ : <<1; inQ (z_n) (x_n)>>; i_t>>%AS <= <<1; w>>%AS)%VS /\ w \in <<1; w>>%AS Hidden 2 goal(s) </pre>
<pre> apply/esym/eqP; rewrite eqEsubv adjoinSl ?sub1v </pre>	

// =; Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
apply/FadjoinP.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 </pre> <hr/> <p style="text-align: center;"><i>exists2 m : nat, (n <= m)%N & extRz m</i> Hidden 1 goal(s)</p>

by rewrite

memv_adjoin / =

defQw Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-galois_connection.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 </pre> <hr/> <pre> have nz2 : 2% : R != 0 : > Qt by move/char f0P : (charQ (Q t)) => </pre>
	<pre> exists2 m : nat, (n <= m)%N & extRz m Hidden 1 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
->.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 (forall w0 : Qt, w0 \in Rz -> w0 \notin Cn /\ w0 ^ + 2 \in Cn -> exists2 m : nat, (n <= m)%N & extRz m) -> exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>
<p>without loss{deg_w} [Ctw Cw2] : w Rz_w / w</p>	

\notin Cn /\ w ^ + 2 \in Cn. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt (forall w0 : Qt, w0 \in Rz -> w0 \notin Cn /\ w0 ^ 2 \in Cn -> exists2 m : nat, (n <= m)%N & extRz m) -> exists2 m : nat, (n <= m)%N & extRz m </pre>
pose p := minPoly Cn w; pose v := p`_1 / 2% : R.	Hidden 2 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn (forall w0 : Qt, w0 \in Rz -> w0 \notin Cn /\ w0 ^ + 2 \in Cn -> exists2 m : nat, (n <= m)%N & extRz m) ->exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>
<p>have /polyOverP Cp : p \is a</p>	

polyOver Cn :=
minPolyOver Cn
w.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn </pre> <hr/> <pre> (forall w0 : Qt, w0 \in Rz -> w0 \notin Cn /\ w0 ^ 2 \in Cn -> exists2 m : nat, (n <= m)%N & extRz m) -> exists2 m : nat, (n <= m)%N & extRz m </pre>
have Cv : v \in Cn by rewrite	Hidden 2 goal(s)

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn </pre> <hr/> <p style="text-align: center;">$v + w \notin Cn \wedge (v + w)^2 \in Cn$</p>

move/(_ (v + w));

apply; first by
rewrite rpredD //
subvP_adjoin.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn </pre> <hr/> <p style="text-align: center;">(v + w) ^ + 2 \in Cn Hidden 2 goal(s)</p>

split; first by

rewrite rpredDt //
-adjoin_deg_eq1

deg_w.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn </pre> <hr/> <p style="text-align: center;">(w + v) ^ + 2 - p.[w] \in Cn Hidden 2 goal(s)</p>

rewrite addrC -[_
^ + 2]subr0 = (rootP
(root_minPoly Cn
w)) -/p.

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn </pre> <hr/> <pre> w ^ + 2 + w * p`_1 - p.[w] \in Cn Hidden 2 goal(s) </pre>
rewrite sqrrD [_ - _]addrAC rpredD	

?rpredX Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-mulr_natr
-mulrA divfK //.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn \sum_(i0 < size p) p`_i0 * w ^ i0 - (p`_1 * w + w ^ 2) \in Cn Hidden 2 goal(s) </pre>

rewrite [w ^ 2 +

_]addrC Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
-rpredN opprB
horner_coef.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G w : Qt Rz_w : w \in Rz deg_w : adjoin_degree Cn w = 2 nz2 : 2% : R != 0 p := minPoly Cn w : {poly Qt} v := p`_1 / 2% : R : Qt Cp : forall i : nat, p`_i \in Cn Cv : v \in Cn (minPoly Cn w)`_3. - 1 = 1 -> \sum_(i0 < 3) p`_i0 * w ^ i0 - (p`_1 * w + w ^ 2) \in Cn Hidden 2 goal(s) </pre>
<pre> have /monicP := monic_minPoly Cn w; rewrite </pre>	

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ + 2 \in Cn </pre> <hr/> <p><i>exists2 m : nat, (n <= m)%N & extRz m</i></p>
<i>by rewrite</i> <i>2!big_ord_recl</i>	Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
big_ord1
rewrite mulr1
mul1r addrK Cp

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ + 2 \in Cn (forall w0 : Qt, w0 \in Rz -> w0 \notin Cn -> w0 ^ + 2 \in Cn -> w0 ^ + 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m) ->exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>
without loss <i>Rw2</i> : <i>w Rz_w Cnw Cw2 /</i>	

w ^ + 2 \notin Rn Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z <i>Rz</i> := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz <i>G</i> := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ + 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ + 2 \in Cn -> w ^ + 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ + 2 \in Rn exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s) </pre>

move = ~~Cdiff~~ Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
have [Rw2 | /IHw]
:= boolP (w ^ + 2
\in Rn); last exact.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ + 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ + 2 \in Cn -> w ^ + 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ + 2 \in Rn Rit : i_t \notin Rn </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m</p> <p style="text-align: center;">Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
have *Rrit* : *i_t* \notin Rn by
rewrite *memRn*
Dit.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ 2 \in Cn -> w ^ 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ 2 \in Rn Rit : i_t \notin Rn v := 1 + i_t : Qt Rv : v \notin Rn </pre> <hr/> <p>exists2 m : nat, (n <= m)%N & extRz m</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
pose v := 1 + i_t,
have Rv : v \notin
Rn by rewrite
rpredDl ?rpred1.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ 2 \in Cn -> w ^ 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ 2 \in Rn Rvit : i_t \notin Rn v := 1 + i_t : Qt Rv : v \notin Rn Cv : v \in Cn </pre> <hr/> <p>exists2 m : nat, (n <= m)%N & extRz m</p> <p>Hidden 2 goal(s)</p>

have Cv Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
by rewrite rpredD
?rpred1
?memv_adjoin

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ 2 \in Cn -> w ^ 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ 2 \in Rn Rit : i_t \notin Rn v := 1 + i_t : Qt Rv : v \notin Rn Cv : v \in Cn 377 nz_v : v != 0 </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m Hidden 2 goal(s)</p>

have nz_v : v != 0
by rewrite
(memPnC Rv)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ + 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ + 2 \in Cn -> w ^ + 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ + 2 \in Rn Rvit : i_t \notin Rn v := 1 + i_t : Qt Rv : v \notin Rn Cv : v \in Cn 378 nz_v : v != 0 </pre> <hr/> <p style="text-align: center;"> $(v * w) ^ + 2 \in Cn$ Hidden 3 goal(s) </p>

apply : (*IHw* (*v* *
w)); last 1 [] [] by
rewrite fpredMl //

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ 2 \in Cn -> w ^ 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ 2 \in Rn Rit : i_t \notin Rn v := 1 + i_t : Qt Rv : v \notin Rn Cv : v \in Cn 379 nz_v : v != 0 </pre> <hr/> <p>(v * w) ^ 2 \notin Rn Hidden 2 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
by rewrite exprMn
rpredM // rpredX.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ 2 \in Cn <i>IHw</i> : forall w : Qt, w \in Rz -> w \notin Cn -> w ^ 2 \in Cn -> w ^ 2 \notin Rn -> exists2 m : nat, (n <= m)%N & extRz m Rw2 : w ^ 2 \in Rn Rit : i_t \notin Rn v := 1 + i_t : Qt Rv : v \notin Rn Cv : v \in Cn 380 nz_v : v != 0 </pre> <hr/> <pre> v ^ 2 \notin (CnQt @ : <<1; inQ (z_n) (x_n)>>%VS </pre>
rewrite exprMn	Hidden 2 goal(s)

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2 - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ 2 \in Cn Rtw2 : w ^ 2 \notin Rn </pre> <hr/> <p><i>exists2 m : nat, (n <= m)%N & extRz m</i> Hidden 1 goal(s)</p>
<p>by rewrite sqrrD Dit2 expr1n addrC</p>	

addKr -Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
fpredMl
?rpred_nat.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ 2 \in Cn Rnw2 : w ^ 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ 2] : Qt -> Qt -> Prop </pre> <hr/> <p style="text-align: center;">exists2 m : nat, (n <= m)%N & extRz m</p> <p style="text-align: center;">Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

pose rect_w2 u v :=
[/\ u \in Rn, v \in
Rn & u + i_t * (v *
2% : R) = w ^ 2].

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ + 2 \in Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop </pre> <hr/> <p style="text-align: center;"> $\{u : Qt \ \& \ \{v : Qt \mid rect_w2 \ u \ v\}\}$ Hidden 2 goal(s) </p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
have{Cw2} {u [v
[Ru Rv Dw2]]] : {u :
Qt & {v | rect_w2
u v}}.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ 2 \in Cn Rnw2 : w ^ 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/ \ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ 2] : Qt -> Qt -> Prop p := Fadjoin_poly Rn i_t (w ^ 2) : {poly Qt} {u : Qt & {v : Qt [/ \ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = p.[i_t]}} Hidden 2 goal(s) </pre>

rewrite /rect_w2; continue proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 -(Fadjoin_poly_eq
 Cw2); set p :=
 Fadjoin_poly Rn
 i_t _.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ + 2 \in Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop p := Fadjoin_poly Rn i_t (w ^ + 2) : {poly Qt} Rp : forall i : nat, p`_i \in Rn </pre> <hr/> <pre> {u : Qt & {v : Qt [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = p.[i_t]}} </pre>

have /polyOver. Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
Rp : p \is a
polyOver Rn by
apply :
Fadjoin_polyOver.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Cw2 : w ^ + 2 \in Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/ \ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop p := Fadjoin_poly Rn i_t (w ^ + 2) : {poly Qt} Rp : forall i : nat, p`_i \in Rn </pre> <hr/> <p style="text-align: center;"> $p_0 + i_t * (p_1 / 2\% : R * 2\% : R) = p.[i_t]$ Hidden 2 goal(s) </p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
 exists p`_0, (p`_1
 / 2% : R); split;
 rewrite ?rpred_div
 ?rpred_nat //.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Cw2 : w ^ + 2 \in Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop p := Fadjoin_poly Rn i_t (w ^ + 2) : {poly Qt} Rp : forall i : nat, p`_i \in Rn p`_0 + i_t * p`_1 = \sum_(i0 < adjoin_degree Rn i_t) p`_i0 * i_t ^ + i0 </pre>
	387 Hidden 2 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
rewrite (coeff) (horner_coef_wide

(size_Fadjoin_poly
_ _ _)) -/p.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 exists2 m : nat, (n <= m)%N & extRz m </pre>
by rewrite adjoin_degreeE	Hidden 1 goal(s)

dimCn Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
big_ord_recl
big_ord1 mulr1
mulrC.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ 2 p := Poly [:- ofQ t v ^ 2; 0; - ofQ t u; 0; 1] : {poly C} </pre> <hr/> <p>exists2 m : nat, (n <= m)%N & extRz m</p> <p>Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
pose p := Poly [:- ofQ t v ^ 2; 0;
- ofQ t u; 0; 1].

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 390 has_Rroot (xR n) p (ofQ t v) Hidden 2 goal(s) </pre>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page
have $\llbracket m \text{ len } m \rrbracket x$
 $Rx \text{ px0} \rrbracket := xRroot$
 $n \text{ p } (ofQ \text{ t } v)$.

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 391 [&& all (mem (sQ (tag (xR n)))) p, p` (size p). - 1 == 1, true & p`_0 == - ofQ t v ^ + 2] Hidden 2 goal(s) </pre>
<i>rewrite /has_Root</i>	

2!unfold_Coq continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

lead_coefE

hopper_coef0

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t w ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 392 [&& ofQ t v ^ + 2 \in sQ (tag (xR n)), ofQ t u \in sQ (tag (xR n)) & true] &&true Hidden 2 goal(s) </pre>

rewrite (@PolyK

1) ?oner Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

leqxx !rpred0

?rpred1 ?rpredN

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 393 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x </pre> <hr/> <p><i>exists2</i> m0 : nat, (n <= m0)%N & <i>extRz</i> m0</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 394 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x y : C Cy : y \in sQ (z_m) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 395 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x y : C Cy : y \in sQ (z_m) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 396 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x </pre>
	<hr/> $\{y : C \mid y \in sQ (z_m) \text{ \& ofQ } t \text{ } w^+ + 2 == y^+ + 2\}$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 397 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x </pre> <hr/> <p style="text-align: center;">$x + i * (ofQ\ t\ v / x) \in sQ\ (z_m)$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 398 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x </pre>
	<pre> ofQ t v \in sQ (x_m) </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 399 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : root p x </pre> <hr/> <p style="text-align: center;">$ofQ\ t\ w\ ^\wedge + 2 == (x + i * (ofQ\ t\ v / x))\ ^\wedge + 2$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 400 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : ((0 + 1) * x * x - ofQ t u) * x * x - ofQ t v ^ + 2 == 0 </pre> <hr/> <p style="text-align: center;">$ofQ\ t\ w\ ^\wedge + 2 == (x + i * (ofQ\ t\ v / x))\ ^\wedge + 2$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 401 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 </pre> <hr/> <p style="text-align: center;">$ofQ\ t\ w\ ^ + 2 == (x + i * (ofQ\ t\ v\ /\ x))\ ^ + 2$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Rtw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 402 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 </pre> <hr/> <p style="text-align: center;">$x ^ + 2 != 0$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} m : nat knm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 y2_0 : x ^ + 2 = 0 </pre> <hr/> <p style="text-align: center;">$u + v * (i_t * 2\% : R) \in Rn$</p>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := <<Cn; u_z>>%AS : {subfield Qt} le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} m : nat 404denm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 y2_0 : x ^ + 2 = 0 </pre>
	$v == 0$

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 405 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 nz_x2 : x ^ + 2 != 0 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 406 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 nz_x2 : x ^ + 2 != 0 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cw : w \notin Cn Rtw2 : w ^ 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ 2 p := Poly [:- ofQ t v ^ 2; 0; - ofQ t u; 0; 1] : {poly C} 407 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ 2 - ofQ t u) * x ^ 2 == ofQ t v ^ 2 nz_x2 : x ^ 2 != 0 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	<pre> <i>extendsR</i> := fun xR yR : realC => tag xR \in sQ (tag yR) : realC -> realC -> bool z : C z_i : i \in sQ z d : nat <i>IHd</i> : forall (p : {poly C}) (n : nat), (size p < d)%N -> [&& p \is monic, root p z & p \is a polyOver (sQ (z_n))] -> exists x : nat_choiceType, (z \in sQ (z_x)) = true n : nat t : C t_C : z_n \in sQ t t_z : z \in sQ t gal_t : is_Gal t <i>Qt</i> := SplittingFieldType rat (Q t) gal_t : splittingFieldType rat_fieldType CnQt : !AHom(Q (z_n), Q t) CnQtE : morph_ofQ (z_n) t CnQt <i>Rn</i> := ((CnQt @ : R_n)%AS : {subfield Qt}) : {subfield Qt} i_t := (CnQt (i_n) : Qt) : Qt Cn := (<<Rn; i_t>>%AS : {subfield Qt}) : {subfield Qt} defCn : Cn = limg CnQt <i>memRn</i> : forall u : Qt, (u \in Rn) = (ofQ t u \in sQ (x_n)) <i>memCn</i> : forall u : Qt, (u \in Cn) = (ofQ t u \in sQ (z_n)) Dit : ofQ t i_t = i Dit2 : i_t ^ + 2 = -1 dimCn : \dim_Rn Cn = 2 u_z : Q t Dz : ofQ t u_z = z Rz := (<<Cn; u_z>>%AS : {subfield Qt}) le_Rz_d : (\dim_Cn Rz < d)%N sRCn : (Rn <= Cn)%VS sCnRz : (Cn <= Rz)%VS sRnRz : (Rn <= Rz)%VS galRz : galois Rn Rz galCz : galois Cn Rz G := (!Gal(Rz / Cn))%G : group_type (gal_finGroupType Rz) ntG : G != 1%g <i>extRz</i> := fun m : nat => exists2 w : Q t, ofQ t w \in sQ (z_m) & w \in [predD Rz & Cn] : nat -> Prop evenG : 2. - group G nz2 : 2% : R != 0 w : Qt Rz_w : w \in Rz Cnw : w \notin Cn Rnw2 : w ^ + 2 \notin Rn <i>rect_w2</i> := fun u v : Qt => [/\ u \in Rn, v \in Rn & u + i_t * (v * 2% : R) = w ^ + 2] : Qt -> Qt -> Prop u : Qt v : Qt Ru : u \in Rn Rv : v \in Rn Dw2 : u + i_t * (v * 2% : R) = w ^ + 2 p := Poly [:- ofQ t v ^ + 2; 0; - ofQ t u; 0; 1] : {poly C} 408 m : nat lenm : (n <= m)%N x : C Rx : x \in sQ (tag (xR m)) px0 : (x ^ + 2 - ofQ t u) * x ^ + 2 == ofQ t v ^ + 2 nz_x2 : x ^ + 2 != 0 </pre>

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$by\ rewrite$ $-mulrnAl$ $-mulrnAr$ $-rmorphMn$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n_ : C \rightarrow nat$ $\boxed{FTA} : forall\ x : C, x \setminus in\ sQ\ (z_ (n_ x))$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$
$-!mulrDl\ addrAC$ $subrK.$ $have\ inFTA\ n\ z :$ $(n_ z <= n)\%N$ $\rightarrow z = ofQ\ (z_ n)$ $(inQ\ (z_ n)\ z).$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ $z : C$ <hr/> $(n_ z <= n)\%N \rightarrow z = ofQ\ (z_ n)\ (inQ\ (z_ n)\ z)$ Hidden 1 goal(s)
$by\ move/sCle =>$ $le_zn;$ $rewrite$ $inQ_K\ ?le_zn.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $\boxed{inFTA} : forall\ (n : nat)\ (z : C), (n_ z <= n)\%N \rightarrow z = ofQ\ (z_ n)\ (inQ\ (z_ n)\ z)$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$
	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $\boxed{is_cj} := fun\ (n : nat)\ (cj : Q\ (z_ n) \rightarrow Q\ (z_ n)) => \{in\ R_ n, cj = 1\ id\} \wedge cj\ (i_ n) = -\ i_ n : forall\ n : nat,$ $(Q\ (z_ n) \rightarrow Q\ (z_ n)) \rightarrow$ $Prop$ <hr/> $\{conj : \{rmorphism\ C \rightarrow C\} \mid involutive\ conj \ \& \ \sim\ conj = 1\ id\}$
$have\ /all_sig[cj_$ $/all_and2[cj_R$ $cj_i]]\ n : \{cj :$ $PAE\ n\ (Q\ (z_ n)\ cj) =$ $\{in\ R_ n, cj = 1\ id\} \wedge$ $cj\ (i_ n) = -\ i_ n.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ <hr/> $\{cj : \iota AEnd(Q\ (z_ n)) \mid is_cj\ n\ cj\}$ Hidden 1 goal(s)
$have\ cj_P : root$ $(minPoly\ (R_ n)$ $(i_ n) ^ \setminus 1\%VF)$ $(- i_ n).$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ <hr/> $root\ (minPoly\ (R_ n)\ (i_ n) ^ \setminus 1\%VF)\ (- i_ n)$ Hidden 2 goal(s)
$rewrite\ minp_i$ $-(fmorph_root$ $(ofQ\ _))\ !rmorphD$ $!rmorph1\ /=$ $!map_polyXn.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n : nat$ <hr/> $root\ (\iota X^2 + 1)\ (ofQ\ (z_ n)\ (- i_ n))$ Hidden 2 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
<i>by rewrite rmorphN inQ_K // rootE hornerD hornerXn hornerC</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $cj_P : \text{root } (\text{minPoly } (R_n) (i_n) \wedge \setminus 1\%VF) (-i_n)$ <hr/> $\{cj : \text{!AEnd}(Q(z_n)) \mid \text{is_cj } n \text{ } cj\}$ <p>Hidden 1 goal(s)</p>
<i>sqrrN Di2 addNr.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $cj_P : \text{root } (\text{minPoly } (R_n) (i_n) \wedge \setminus 1\%VF) (-i_n)$ <hr/> $\text{ahom_in fullv } (k\text{HomExtend } (R_n) \setminus 1 (i_n) (-i_n))$ <p>Hidden 2 goal(s)</p>
<i>have cj_M : ahom_in fullv (kHomExtend (R_n) n) \setminus 1 (i_n) (-i_n)).</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $cj_P : \text{root } (\text{minPoly } (R_n) (i_n) \wedge \setminus 1\%VF) (-i_n)$ $cj_M : \text{ahom_in fullv } (k\text{HomExtend } (R_n) \setminus 1 (i_n) (-i_n))$ <hr/> $\{cj : \text{!AEnd}(Q(z_n)) \mid \text{is_cj } n \text{ } cj\}$ <p>Hidden 1 goal(s)</p>
<i>by rewrite -def Ri -k1HomE kHomExtendP ?sublv ?kHom1.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $n : \text{nat}$ $cj_P : \text{root } (\text{minPoly } (R_n) (i_n) \wedge \setminus 1\%VF) (-i_n)$ $cj_M : \text{ahom_in fullv } (k\text{HomExtend } (R_n) \setminus 1 (i_n) (-i_n))$ <hr/> $\text{AHom } cj_M (i_n) = -i_n$ <p>Hidden 1 goal(s)</p>
<i>exists (AHom cj_M); split => [y /kHomExtend_id->]; first by rewrite ?id_lfunE.</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $\boxed{cj_} : \text{forall } x : \text{nat}, \text{!AEnd}(Q(z_x))$ $\boxed{cj_R} : \text{forall } x : \text{nat}, \{in R_x, cj_x = 1 \text{ id}\}$ $\boxed{cj_i} : \text{forall } x : \text{nat}, cj_x (i_x) = -i_x$ <hr/> $\{conj : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive } conj \ \& \ \sim conj = 1 \text{ id}\}$
<i>by rewrite (kHomExtend_val (kHom1 1 _)).</i>	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $\boxed{conj_} := \text{fun } (n : \text{nat}) (z : C) => \text{ofQ } (z_n) (cj_n (\text{inQ } (z_n) z)) : \text{nat} \rightarrow C \rightarrow C$ $\boxed{conj} := \text{fun } z : C => conj_ (n_z) z : C \rightarrow C$ <hr/> $\{conj0 : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive } conj0 \ \& \ \sim conj0 = 1 \text{ id}\}$
<i>pose conj_n z := ofQ - (cj_n (inQ _ z)); pose conj z := conj_n z.</i>	

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\begin{aligned} & \text{have conjK } n \ m \ z : \\ & (n_z \leq n) \% N \\ & \rightarrow (n \leq m) \% N \\ & \rightarrow \text{conj_} m \\ & (\text{conj_} n \ z) = z. \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \\ & \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n, m : \text{nat} \\ & \quad z : C \end{aligned}$ <hr/> $(n_z \leq n) \% N \rightarrow (n \leq m) \% N \rightarrow \text{conj_} m \ (\text{conj_} n \ z) = z$ <p>Hidden 1 goal(s)</p>
$\begin{aligned} & \text{move/sCle} \Rightarrow \\ & le_z_n \ le_n_m; \\ & \text{have} \\ & /le_z_n/sQ_inQ[u \\ & \leq] := \text{FTA } z. \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \\ & \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n, m : \text{nat} \\ & \quad z : C \\ & \quad le_z_n : \{\text{subset } sQ \ (z_ (n_z)) \leq sQ \ (z_ n)\} \\ & \quad le_n_m : (n \leq m) \% N \\ & \quad u : Q \ (z_ n) \end{aligned}$ <hr/> $\text{conj_} m \ (\text{conj_} n \ (\text{ofQ} \ (z_ n) \ u)) = \text{ofQ} \ (z_ n) \ u$ <p>Hidden 1 goal(s)</p>
$\begin{aligned} & \text{have } /QtoQ[Qmn \\ & QmnE] : z_n \setminus \text{in} \\ & sQ \ (z_ m) \text{ by} \\ & \text{rewrite } (sCle \ n). \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \\ & \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n, m : \text{nat} \\ & \quad z : C \\ & \quad le_z_n : \{\text{subset } sQ \ (z_ (n_z)) \leq sQ \ (z_ n)\} \\ & \quad le_n_m : (n \leq m) \% N \\ & \quad u : Q \ (z_ n) \\ & \quad Qmn : !AHom(Q \ (z_ n), Q \ (z_ m)) \\ & \quad QmnE : \text{morph_ofQ} \ (z_ n) \ (z_ m) \ Qmn \end{aligned}$ <hr/> $\text{conj_} m \ (\text{conj_} n \ (\text{ofQ} \ (z_ n) \ u)) = \text{ofQ} \ (z_ n) \ u$ <p>Hidden 1 goal(s)</p>
$\begin{aligned} & \text{rewrite } /conj_ \\ & \text{ofQ_K} \ !QmnE \\ & !\text{ofQ_K} \\ & \text{--!comp_lfunE;} \\ & \text{congr } (\text{ofQ_K}) \end{aligned}$	$\begin{aligned} & \text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \\ & \rightarrow \text{realC} \rightarrow \text{bool} \\ & \quad n, m : \text{nat} \\ & \quad z : C \\ & \quad le_z_n : \{\text{subset } sQ \ (z_ (n_z)) \leq sQ \ (z_ n)\} \\ & \quad le_n_m : (n \leq m) \% N \\ & \quad u : Q \ (z_ n) \\ & \quad Qmn : !AHom(Q \ (z_ n), Q \ (z_ m)) \\ & \quad QmnE : \text{morph_ofQ} \ (z_ n) \ (z_ m) \ Qmn \end{aligned}$ <hr/> $((cj_ m \setminus o \ Qmn) \setminus o \ cj_ n) \% VF \ u = Qmn \ u$ <p>Hidden 1 goal(s)</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$move : u (memRi\ n\ u);$ $apply/eqlfun_inP/FadjoinP;$ $split => / = .$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n, m : nat$ $z : C$ $le_z_n : \{subset\ sQ\ (z_ (n_ z)) \leq sQ\ (z_ n)\}$ $le_n_m : (n \leq m) \% N$ $Qmn : \iota AHom(Q\ (z_ n), Q\ (z_ m))$ $QmnE : morph_ofQ\ (z_ n)\ (z_ m)\ Qmn$ <hr/> $(< < 1; inQ\ (z_ n)\ (x_ n) >> \leq lker\ (((cj_ m \setminus o\ Qmn) \setminus o\ cj_ n) \% VF - Qmn)) \% VS$ Hidden 2 goal(s)
$apply/eqlfun_inP =$ $> y\ Ry; rewrite$ $!comp_lfunE$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n, m : nat$ $z : C$ $le_z_n : \{subset\ sQ\ (z_ (n_ z)) \leq sQ\ (z_ n)\}$ $le_n_m : (n \leq m) \% N$ $Qmn : \iota AHom(Q\ (z_ n), Q\ (z_ m))$ $QmnE : morph_ofQ\ (z_ n)\ (z_ m)\ Qmn$ $y : Q\ (z_ n)$ $Ry : y \setminus in\ < < 1; inQ\ (z_ n)\ (x_ n) >>$ <hr/> $Qmn\ y \setminus in\ R_ m$ Hidden 2 goal(s)
$!cj_R //.$ $by\ move : Ry;$ $rewrite\ -!sQof2$ $QmnE\ !inQ\ K //;$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n, m : nat$ $z : C$ $le_z_n : \{subset\ sQ\ (z_ (n_ z)) \leq sQ\ (z_ n)\}$ $le_n_m : (n \leq m) \% N$ $Qmn : \iota AHom(Q\ (z_ n), Q\ (z_ m))$ $QmnE : morph_ofQ\ (z_ n)\ (z_ m)\ Qmn$ <hr/> $i_ n \setminus in\ lker\ (((cj_ m \setminus o\ Qmn) \setminus o\ cj_ n) \% VF - Qmn)$ Hidden 1 goal(s)
$apply : sRle.$ $apply/eqlfunP;$ $rewrite$ $!comp_lfunE\ cj_ i$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $\rightarrow realC \rightarrow bool$ $n, m : nat$ $z : C$ $le_z_n : \{subset\ sQ\ (z_ (n_ z)) \leq sQ\ (z_ n)\}$ $le_n_m : (n \leq m) \% N$ $Qmn : \iota AHom(Q\ (z_ n), Q\ (z_ m))$ $QmnE : morph_ofQ\ (z_ n)\ (z_ m)\ Qmn$ <hr/> $- cj_ m\ (Qmn\ (i_ n)) = Qmn\ (i_ n)$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$suffices -> : Qmn$ $(i_n) = i_m$ by $rewrite\ cj_i$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $-> realC -> bool$ $n, m : nat$ $z : C$ $le_z_n : \{subset\ sQ\ (z_ (n_ z)) <= sQ\ (z_ n)\}$ $le_n_m : (n <= m) \% N$ $Qmn : \iota AHom(Q\ (z_ n), Q\ (z_ m))$ $QmnE : morph_ofQ\ (z_ n)\ (z_ m)\ Qmn$ <hr/> $Qmn\ (i_n) = i_m$ Hidden 1 goal(s)
$?opprK.$ $by\ apply :$ $(fmorph_inj\ (ofQ_)) ; rewrite$ $QmnE\ !inQ_K.$ $have\ conjE\ n\ z :$ $(n_ z <= n) \% N$ $-> conj\ z = conj_n\ z.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $-> realC -> bool$ $\boxed{conjK} : forall\ (n\ m : nat)\ (z : C),\ (n_ z <= n) \% N -> (n <= m) \% N -> conj_m\ (conj_n\ z) = z$ <hr/> $\{conj0 : \{rmorphism\ C -> C\} \mid involutive\ conj0 \ \&\ \sim\ conj0 = 1\ id\}$
$move/leq_trans =$ $> le_zn ; set\ x :=$ $conj\ z ; set\ y :=$ $conj_n\ z.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $-> realC -> bool$ $n : nat$ $z : C$ $le_zn : forall\ n0 : nat,\ (n <= n0) \% N -> (n_ z <= n0) \% N$ $x := conj\ z : C$ $y := conj_n\ z : C$ <hr/> $x = y$ Hidden 1 goal(s)
$have\ [m\ [le_xm$ $le_ym\ le_nm]] :=$ $maxn3\ (n_x)\ (n_y)\ n.$	$extendsR := fun\ xR\ yR : realC => tag\ xR \setminus in\ sQ\ (tag\ yR) : realC$ $-> realC -> bool$ $n : nat$ $z : C$ $le_zn : forall\ n0 : nat,\ (n <= n0) \% N -> (n_ z <= n0) \% N$ $x := conj\ z : C$ $y := conj_n\ z : C$ $m : nat$ $le_xm : (n_x <= m) \% N$ $le_ym : (n_y <= m) \% N$ $le_nm : (n <= m) \% N$ <hr/> $x = y$ Hidden 1 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
$\text{by have } /conjK/ =$ $/can_in_inj :=$ $leqnn m; \text{ apply;}$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $\boxed{\text{conjE}} : \text{forall } (n : \text{nat}) \ (z : C), (n_ z \leq n) \% N \rightarrow \text{conj } z =$ $\text{conj_ } n \ z$ <hr/> $\{\text{conj0} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj0} \ \& \ \sim \text{conj0} = 1 \text{ id}\}$
rewrite ?conjK // $le_zn.$ suffices conjM : rmorphism conj.	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $\text{conjM} : \text{rmorphism conj}$ <hr/> $\{\text{conj0} : \{\text{rmorphism } C \rightarrow C\} \mid \text{involutive conj0} \ \& \ \sim \text{conj0} = 1 \text{ id}\}$ Hidden 1 goal(s)
exists (RMorphism $\text{conjM}) \Rightarrow [z \mid$ $/(_ i)/eqP/idPn[]]$ $/ = .$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $\text{conjM} : \text{rmorphism conj}$ $z : C$ <hr/> $\text{conj (conj } z) = z$ Hidden 2 goal(s)
by have [n $[/conjE \rightarrow$ $/(conjK (n_ z)) \rightarrow]] := \text{maxn3}$ $(n_ (conj z)) (n_ z) \ 0 \% N.$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $\text{conjM} : \text{rmorphism conj}$ <hr/> $\text{conj } i \neq i$ Hidden 1 goal(s)
rewrite $/conj/conj_cj_i$ rmorphN inQ_K $// \text{eq_sym}$ $-addr_eq0$ $-mulr2n$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $\text{conjM} : \text{rmorphism conj}$ <hr/> $2\% : R * i \neq 0$ Hidden 1 goal(s)
$-mulr_natl.$ rewrite mulf_neq0 $?(memPnC (R/i$ $0 \% N)) ?rpred0 //.$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $\text{conjM} : \text{rmorphism conj}$ <hr/> $2\% : R \neq 0$ Hidden 1 goal(s)
by have $/charf0P \rightarrow :=$ ftrans $(\text{fmorph_char}$ $\text{QtoC}) (\text{char_num}$ $_).$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ <hr/> rmorphism conj
$\text{do } 2?split \Rightarrow [x$ $y]; \text{ last pose n1} :=$ $n_1.$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC}$ $\rightarrow \text{realC} \rightarrow \text{bool}$ $x, y : C$ <hr/> $\text{conj } (x - y) = \text{conj } x - \text{conj } y$ Hidden 2 goal(s)

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
—	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $x, y : C$ <hr/> $\text{conj } (x - y) = \text{conj } x - \text{conj } y$ <p>Hidden 2 goal(s)</p>
$\text{have } [m \ [\text{le_xm} \ \text{le_ym} \ \text{le_xym}]] := \text{maxn3 } (n_x) (n_y) (n_ (x - y)).$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $x, y : C$ $m : \text{nat}$ $\text{le_xm} : (n_x \leq m) \% N$ $\text{le_ym} : (n_y \leq m) \% N$ $\text{le_xym} : (n_ (x - y) \% R \leq m) \% N$ <hr/> $\text{conj } (x - y) = \text{conj } x - \text{conj } y$ <p>Hidden 2 goal(s)</p>
$\text{by rewrite } !(\text{conjE } m) // (\text{inFTA } m \ x) // (\text{inFTA } m \ y) -? \text{rmorphB} / \text{conj_ofQ_K}.$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $x, y : C$ <hr/> $\text{conj } (x * y) = \text{conj } x * \text{conj } y$ <p>Hidden 1 goal(s)</p>
—	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $x, y : C$ <hr/> $\text{conj } (x * y) = \text{conj } x * \text{conj } y$ <p>Hidden 1 goal(s)</p>
$\text{have } [m \ [\text{le_xm} \ \text{le_ym} \ \text{le_xym}]] := \text{maxn3 } (n_x) (n_y) (n_ (x * y)).$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $x, y : C$ $m : \text{nat}$ $\text{le_xm} : (n_x \leq m) \% N$ $\text{le_ym} : (n_y \leq m) \% N$ $\text{le_xym} : (n_ (x * y) \% R \leq m) \% N$ <hr/> $\text{conj } (x * y) = \text{conj } x * \text{conj } y$ <p>Hidden 1 goal(s)</p>
$\text{by rewrite } !(\text{conjE } m) // (\text{inFTA } m \ x) // (\text{inFTA } m \ y) -? \text{rmorphM}$	$\text{extendsR} := \text{fun } xR \ yR : \text{realC} \Rightarrow \text{tag } xR \setminus \text{in } sQ \ (\text{tag } yR) : \text{realC} \rightarrow \text{realC} \rightarrow \text{bool}$ $\boxed{n1} := n_1 : \text{nat}$ <hr/> $\text{conj } 1 = 1$
$\text{by rewrite } !(\text{conjE } m) // (\text{inFTA } m \ x) // (\text{inFTA } m \ y) -? \text{rmorphM}$	<p>Proof finished by Qed</p>

Continuing proof of Theorem Fundamental_Theorem_of_Algebraics on the next page

Table 1: Proof of Theorem Fundamental_Theorem_of_Algebraics
continued

Next step in Coq	Proof situation
	End of proof of Theorem Fundamental_Theorem_of_Algebraics