See the proof for Theorem nat_to_binary_is_zero in table 1

Table 1: Proof of Theorem nat_to_binary_is_zero

Next step in Coq	Proof situation
Proof.	$\vdash$ *forall i : nat, binary_is_zero (nat_to_binary i) −> i = 0*
intro i.	$\boxed{\text{i}}$ *: nat* $\vdash$ *binary_is_zero (nat_to_binary i) −> i = 0*
pattern i.	$\vdash$ *(fun n : nat = > binary_is_zero (nat_to_binary n) −> n = 0) i*
apply binary_induction; [tauto| |].	$\vdash$ *forall n : nat, n <> 0 −>(binary_is_zero (nat_to_binary n) −> n = 0) −>binary_is_zero (nat_to_binary (2 ∗ n)) −> 2 ∗ n = 0* Hidden 1 goal(s)
intros n H1 IH H2.	*n : nat ; H1 : n <> 0 ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ; H2 : binary_is_zero (nat_to_binary (2 ∗ n))* $\vdash$ *2 ∗ n = 0* Hidden 1 goal(s)
rewrite IH; [tauto|].	*n : nat ; H1 : n <> 0 ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ; H2 : binary_is_zero (nat_to_binary (2 ∗ n))* $\vdash$ *binary_is_zero (nat_to_binary n)* Hidden 1 goal(s)
rewrite nat_to_binary_even in H2; [|exact H1].	*n : nat ; H1 : n <> 0 ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ; H2 : binary_is_zero (false :: nat_to_binary n)* $\vdash$ *binary_is_zero (nat_to_binary n)* Hidden 1 goal(s)
inversion H2.	*n : nat ; H1 : n <> 0 ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ; H2 : binary_is_zero (false :: nat_to_binary n) ; x : bool ; l : list bool ; H3 : false = false ; H4 : Forall (eq false) (nat_to_binary n) ; H : x = false ; H0 : l = nat_to_binary n* $\vdash$ *binary_is_zero (nat_to_binary n)* Hidden 1 goal(s)
assumption.	$\vdash$ *forall n : nat, (binary_is_zero (nat_to_binary n) −> n = 0) −>binary_is_zero (nat_to_binary (2 ∗ n + 1)) −> 2 ∗ n + 1 = 0*
intros n IH H1.	*n : nat ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ; H1 : binary_is_zero (nat_to_binary (2 ∗ n + 1))* $\vdash$ *2 ∗ n + 1 = 0*
rewrite nat_to_binary_odd in H1.	*n : nat ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ;* $\boxed{\text{H1}}$ *: binary_is_zero (true :: nat_to_binary n)* $\vdash$ *2 ∗ n + 1 = 0*
inversion H1.	*n : nat ; IH : binary_is_zero (nat_to_binary n) −> n = 0 ; x : bool ; l : list bool ;* $\boxed{\text{H2}}$ *: false = true ;* $\boxed{\text{H3}}$ *: Forall (eq false) (nat_to_binary n) ;* $\boxed{\text{H}}$ *: x = true ; H0 : l = nat_to_binary n* $\vdash$ *2 ∗ n + 1 = 0*
discriminate.	Proof finished by Qed

End of proof of Theorem nat_to_binary_is_zero