Porting the Mathematical Components library to Hierarchy Builder

MC^{HB}

Reynald Affeldt, Xavier Allamigeon, Yves Bertot, Quentin Canu, Cyril Cohen, Pierre Roux, Kazuhiko Sakaguchi, Enrico Tassi, Laurent Théry, and Anton Trunov

The Coding Sprint

Context:

- **MC** has a **large hierarchy** of **interfaces**, coded **by hand**, hard to change
- **HB** is a high-level language / tool to declare hierarchies of interfaces for Coq
- porting MC to HB was a **daunting task** for the 3 authors of HB

To the rescue:

estimate 400 hours!!!

Tools for the sprint

Process wiki/HB-porting-week (knowledge sharing / team building)

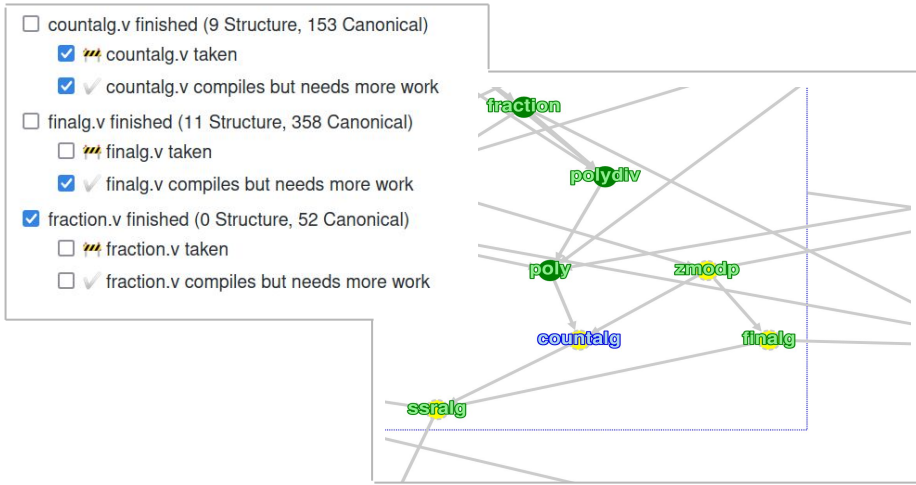
- Demo all together
- Initiation/setup in large groups, passing the baton
- Smaller and smaller groups tackling files
- opam pinning / nix toolbox to update HB frequently

Code & synchronization #733

- Library graph superposed with status

Chat coq.zulipchat.com + jitsi.riot.im

- each file had a Zulip stream
- and a video chat with standard names (easy to jump to)



The diff

727c50f

```
gares@ollypat:~/MATHCOMP/math-comp$ git diff master..hierarchy-builder --stat mathcomp
 mathcomp/Make                            |    2 +
 mathcomp/_CoqProject                     |    1 +
 mathcomp/algebra/countalg.v              |  793 ++++-----------------------------------------
 mathcomp/algebra/finalg.v                | 1707 +++++++++++++++----------------------------------------------------------------------------
 mathcomp/algebra/fraction.v              |   79 +----
 mathcomp/algebra/intdiv.v                |    2 +-
 mathcomp/algebra/interval.v              |  120 ++++---
 mathcomp/algebra/matrix.v                |  157 ++++-----
 mathcomp/algebra/mxalgebra.v             |  190 ++++++-----
 mathcomp/algebra/mxpoly.v                |    6 +-
 mathcomp/algebra/poly.v                  |  140 +++-----
 mathcomp/algebra/polyXY.v                |    1 +
 mathcomp/algebra/rat.v                   |   65 +---
 mathcomp/algebra/ring_quotient.v         |  410 +++++++-----------------
 mathcomp/algebra/ssralg.v                | 2220 +++++++++++++++++++++++++++++++++++++++++++++++++++---------------------------------------------------------
 mathcomp/algebra/ssrint.v                |   61 ++--
 mathcomp/algebra/ssrnum.v                | 1732 ++++++++++++++++++++++++++------------------------------------------------------------------------
 mathcomp/algebra/vector.v                |  244 ++++++--------
 mathcomp/algebra/zmodp.v                 |   65 ++--
 mathcomp/character/character.v           |   40 +--
 mathcomp/character/classfun.v            |   83 +++--
 mathcomp/character/integral_char.v       |   16 +-
 mathcomp/character/mxabelem.v            |   18 +-
 mathcomp/character/mxrepresentation.v    |  132 +++-----
 mathcomp/field/algC.v                    |  318 +++++++++---------
 mathcomp/field/algebraics_fundamentals.v |   25 +-
 mathcomp/field/algnum.v                  |    5 +-
 mathcomp/field/closed_field.v            |   67 ++--
 mathcomp/field/cyclotomic.v              |    2 +-
 mathcomp/field/falgebra.v                |  218 +++++--------
 mathcomp/field/fieldext.v                |  543 ++++++++++---------------------
 mathcomp/field/finfield.v                |  207 ++++++------
 mathcomp/field/galois.v                  |  165 +++-------
 mathcomp/field/separable.v               |   34 +-
 mathcomp/fingroup/action.v               |   10 +-
 mathcomp/fingroup/fingroup.v             |  356 +++++++++------------
 mathcomp/fingroup/gproduct.v             |   40 +--
 mathcomp/fingroup/perm.v                 |  108 +++----
 mathcomp/fingroup/quotient.v             |   22 +-
 mathcomp/solvable/abelian.v              |    5 +-
 mathcomp/solvable/alt.v                  |    9 +-
 mathcomp/solvable/burnside_app.v         |   25 +-
 mathcomp/solvable/center.v               |    6 +-
 mathcomp/solvable/cyclic.v               |    6 +-
 mathcomp/solvable/extraspecial.v         |    8 +-
 mathcomp/solvable/extremal.v             |   50 +--
 mathcomp/solvable/finmodule.v            |   56 ++--
 mathcomp/solvable/gfunctor.v             |    7 +-
 mathcomp/solvable/hall.v                 |    3 +-
 mathcomp/solvable/jordanholder.v         |   39 +--
 mathcomp/solvable/maximal.v              |    5 +-
 mathcomp/solvable/nilpotent.v            |   11 +-
 mathcomp/solvable/pgroup.v               |    1 +
 mathcomp/solvable/primitive_action.v     |    1 +
 mathcomp/solvable/sylow.v                |    1 +
 mathcomp/ssreflect/bigop.v               |   40 ++-
 mathcomp/ssreflect/choice.v              |  365 +++++++++------------
 mathcomp/ssreflect/eqtype.v              |  250 ++++++++-------
 mathcomp/ssreflect/finfun.v              |   55 ++--
 mathcomp/ssreflect/finset.v              |  126 +++-----
 mathcomp/ssreflect/fintype.v             |  337 ++++++++------------
 mathcomp/ssreflect/generic_quotient.v    |  258 ++++++---------
 mathcomp/ssreflect/order.v               | 3263 +++++++++++++++++++++++++++++++++++++++++++++++++++++++----------------------------------------------------------------------------------------------------------------------------
 mathcomp/ssreflect/seq.v                 |    6 +-
 mathcomp/ssreflect/ssrAC.v               |    9 +-
 mathcomp/ssreflect/ssrnat.v              |   10 +-
 mathcomp/ssreflect/tuple.v               |   32 +-
 67 files changed, 4944 insertions(+), 10444 deletions(-)
```

Locking, not only about performance

Always used in MC to **enforce an abstraction** barrier (for the user, and for Coq)

- a new concept defined using old ones
- a new theory which does not require “unfolding” the concept
- an opacity hint to the Coq kernel

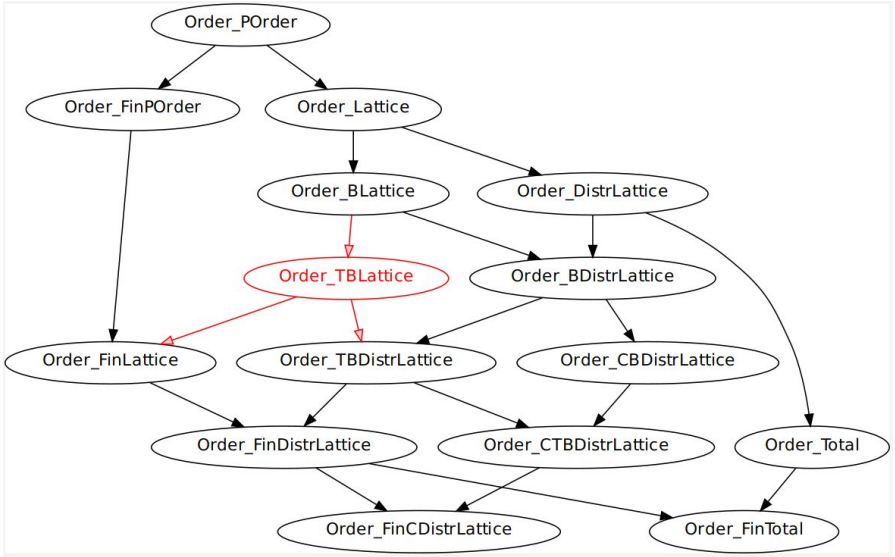
Streamlined with HB:

- HB.lock
  Definition new_concept := …huge… .

```
Module Type new_conceptLocked.
Parameter body : nat.
Parameter unlock : body = ...huge....
End new_conceptLocked.
Module new_concept : new_conceptLocked.
Definition body : nat := ...huge....
Definition unlock : body = ...huge... := eq_refl.
End new_concept.
Notation new_concept :=   new_concept .body
```

Documentation

New tools: HB.graph



HB.about

```
HB: Order.TBLattice.type is a structure (from "./ssreflect/order.v", line 1459)
HB: Order.TBLattice.type characterizing operations and axioms are:
    - lex1
    - top
HB: Order.TBLattice is a factory for the following mixins:
    - choice.HasChoice
    - eqtype.HasDecEq
    - Order.IsPOrdered
    - Order.HasBottom
    - Order.POrder_IsLattice
    - Order.HasTop (* new, not from inheritance *)


HB: Order.HasTop.Build is a factory constructor
    (from "./ssreflect/order.v", line 1450)
HB: Order.HasTop.Build requires its subject to be already equipped with:
    - choice.HasChoice
    - eqtype.HasDecEq
    - Order.IsPOrdered
HB: Order.HasTop.Build provides the following mixins:
    - Order.HasTop
HB: arguments: Order.HasTop.Build d T [top] lex1
    - d : unit
    - T : Type
    - top : T
    - lex1 : forall x : (T), x <= 1
```

Hierarchy Design

HB tries to **detect and forbid** the most tricky **errors** in defining a hierarchy.

But this turned out to be insufficient, since **the users** has still to **fix** his design.

Demo

More doc in the wiki

What Next?

Missing:

- Port the Odd Order Theorem (no new structures, but a huge perf test)
- Add a few more structures, eg semirings (just to test we don't hit a wall)
- Doc, Doc, Doc...

MC^{HB} → MC 2.0, which will be a breaking change - target 2022

- hence MC 1.x will enter maintenance mode (not abandoned)

Related:

- $Finmap^{HB}$, $Monae^{HB}$, $Dioids^{HB}$, $Graph\text{-}Theory^{HB}$, $MC\text{-}Analysis^{HB}$, $CoqEAL^{HB}$, $Multinomials^{HB}$,...

Thanks!

MC^{HB} (PR #733)

HB wiki