

Coq 8.5: what's new, what's next?

M. DÉNÈS[†] & M. SOZEAU^{*}

[†] University of Pennsylvania

^{*} πr^2 - Inria & PPS

Coq POPL Meeting
January 21st 2014
San Diego, USA

- 1 Up and coming
 - Native compilation – M. Dénès
 - Incremental development – E. Tassi
 - Universe polymorphism – M. Sozeau
 - Fast record projections – M. Sozeau
 - New proof engine – A. Spiwack
 - New coqdoc – Y. Régis-Gianas
 - Misc – dev team & contributors
 - The inconsistency
 - Planning
 - opam – T. Braibant
- 2 The close future
 - Performance
 - CoqMT – P.Y. Strub
- 3 The distant future
 - HoTT or OTT?
 - as a programming language

- ▶ NbE style (?)
- ▶ $\text{CoQ} \xrightarrow{\text{nativecomp}} \text{ML} \xrightarrow{\text{ocamlpt}} \text{ASM}$
- ▶ Faster than `vm_compute`

+ *Optional*

++/- *Faster at runtime, compilation is slow*

- ▶ `git` model
- ▶ Granularity: opaque lemmas
- ▶ `.vi` interfaces: cross-file compilation
- ▶ CoqIDE only (emacs interface hard to implement)

-/+++ *Not optional, backwards-compatible*

+++ *Faster interaction and compilation*

- ▶ *Checked* universes.
- ▶ Global definitions (and inductives) or local with prenex, “bounded” quantification. *Conservativity proof*
- ▶ *Elaboration* from typical ambiguity + implicit generalization.
- ▶ Checks HoTT/Coq, B. Barras and T. Coquand’s semi-simplicial model and another (1-)groupoid model.

- ▶ *Checked* universes.
- ▶ Global definitions (and inductives) or local with prenex, “bounded” quantification. *Conservativity proof*
- ▶ *Elaboration* from typical ambiguity + implicit generalization.
- ▶ Checks HoTT/Coq, B. Barras and T. Coquand’s semi-simplicial model and another (1-)groupoid model.
- ▶ Better error messaging, unification sensitive to universes.
- ▶ Generalized rewriting in **Type**

=/++ *Optional except for checked universes, impacts the ML hacker only. Backwards-compatibility layer.*

=+/+ *Comparable or better performance, more expressive*

- ▶ Economic representation
- ▶ Fast reduction
- ▶ No more stupid unfoldings

+ / +− *Optional, backwards-compatibility layer, small source-level incompatibilities.*

+^ω *Exponentially better performance*

TODO: *MS: Graph*

- ▶ Existential variables + LogicT
 - ▶ Proof-search semantics ($;$, $+$, \dots)
 - ▶ Cleaner code: abstract monadic interface
 - ▶ Native `refine`, dependent goals, goal manipulation
- /++ *Not optional, backwards-compatibility layer*
- *0-15% time overhead (still investigating, OCAML issue likely)*

- ▶ Based on the AST and *not* lexing anymore
 - ▶ All semantic information available
 - ▶ Pluggable interface
- + *Not optional but legacy coqdoc code included*
- = *Same performance*

- ▶ Interfaces, documentation, and OCAML best practices (P.M. PÉDROT, ...)
- ▶ Module system simplifications (P. LETOUZEY)
- ▶ Generalized intro patterns (H. HERBELIN, P. LETOUZEY)
- ▶ Better(?) guard condition (P. BOUTILLIER, H. HERBELIN)
- ▶ Rewriting with strategies (M. SOZEAU)

The inconsistency

- ▶ Due to a (very) extensional axiom, related to *equality* on types.
 - ▶ A good fix is known, but breaks the stdlib's use of the smart guard condition (ha ha). All definitions can be adapted. Work by B. BARRAS, C. PAULIN, M. DÉNÈS, P. BOUTILIER, ...
- = Happens in AGDA too!
- ▶ Argues for less syntax and more logic.

- ▶ February 2014: 8.5- α release
- ▶ April 2014: 8.5- β release and opening of new branch (only bug fixes in trunk between α and β)
- ▶ August 2014: final release

- ▶ coq, HoTT/Coq, Coq+SSREFLECT, any version, git included
- ▶ Some user contribs: CONTAINERS, COCCINELLE, ERGO, ...
- ▶ Submit a pull request! Try! Test!

TODO: *TB: USB key?*

The close future

- 1 Up and coming
 - Native compilation – M. Dénès
 - Incremental development – E. Tassi
 - Universe polymorphism – M. Sozeau
 - Fast record projections – M. Sozeau
 - New proof engine – A. Spiwack
 - New coqdoc – Y. Régis-Gianas
 - Misc – dev team & contributors
 - The inconsistency
 - Planning
 - opam – T. Braibant
- 2 The close future
 - Performance
 - CoqMT – P.Y. Strub
- 3 The distant future
 - HoTT or OTT?
 - as a programming language

- ▶ Caching of typechecked terms (E. TASSI, M. SOZEAU)
- ▶ Generalized hashconsing (T. BRAIBANT, M. SOZEAU, P. M. PÉDROT, ...)

- ▶ Waiting for feedback
- ▶ Integrated as an optional feature

The distant future

- 1 Up and coming
 - Native compilation – M. Dénès
 - Incremental development – E. Tassi
 - Universe polymorphism – M. Sozeau
 - Fast record projections – M. Sozeau
 - New proof engine – A. Spiwack
 - New coqdoc – Y. Régis-Gianas
 - Misc – dev team & contributors
 - The inconsistency
 - Planning
 - opam – T. Braibant
- 2 The close future
 - Performance
 - CoqMT – P.Y. Strub
- 3 The distant future
 - HoTT or OTT?
 - as a programming language

HoTT or OTT?

HoTT ideas, math oriented:

- ▶ HITs (prototype by B. BARRAS)
- ▶ Resizing of universes
- ▶ *Definitionally*: $\beta\delta$, $\eta?$, $\iota?$
- ▶ *Propositionally*: proof-irrelevance, functional extensionality, isomorphic types are equal (from an axiom).

OTT/Epigram ideas, CS oriented:

- ▶ Levitation of inductive types
- ▶ Clear Prop/Set separation
- ▶ *Definitionally*: $\beta\delta\eta\iota$, proof-irrelevance, functional extensionality.
- ▶ *Propositionally*: defined, heterogeneous on values, structural on types (\neq iso). With coercion and coherence operators.

Ideally, we should mix the two. . .

Usability, efficiency woes:

- ▶ I/O, FFI (**TODO:** *MD: Demo Hello world! in Coq*)
- ▶ A sliding scale for efficiency/safety in the kernel (idea: tainting features)
- ▶ Definitional proof-irrelevance based on bracket types (useful for DTP and recursion encodings)
- ▶ Type-based termination (sized-types, J. L. SACHINI)