HOL Light: future wishes

John Harrison
Intel Corporation

Workshop on Interactive Theorem Proving
Cambridge

Tue 25th August 2009 (15:00 – 15:15)

What can be improved about HOL Light?

What can be improved about HOL Light?

Nothing, it’s perfect.

What can be improved about HOL Light?

Nothing, it’s perfect.

Questions?

What I don’t want

Not because these are *bad*, but they take us further away from the ideal of simplicity.

- Type classes
- Dependent types
- Abstract theories / modules / locales
- Reflection

What I don’t want

Not because these are *bad*, but they take us further away from the ideal of conceptual simplicity.

- Type classes
- Dependent types
- Abstract theories / modules / locales
- Definitional equality
- Reflection
  - Though internal ‘reflection’ á la Coq *can* be useful and is already used.
  - *Would* be good to have a principled way of doing faster arithmetic.

So what *do* I want?

- System-level improvements
- Proof language improvements
- Infrastructure improvements
- New decision procedures
- Library improvements
- Correctness / proof transfer improvements

System improvements

- Run the system compiled (apparently already possible)
- Save the toplevel OCaml state in a more convenient way.
- Make installation painless for non-programmers.

Proof language improvements

I brought declarative proof into HOL Light back in 1996, but then never used it seriously.

Main tactic language is not much changed since about Cambridge LCF, and is verbose and clumsy.

- Investigate new ways of mixing declarative and procedural proof (‘luxury’ mode).
- Just improve the procedural parts, e.g. more in line with ss-reflect.

Infrastructure improvements

HOL Light already has quite powerful automation in the area of analysis and algebraic reasoning. Less good at things that are useful in classic ‘computer science’ applications.

- Tools for coinductive and corecursive definitions.
- Recursive types involving function spaces built from type being defined.
- Cleverer termination prover for general recursive functions.

New decision procedures

- Simple built-in Nelson-Oppen combination
- Yet more links to external tools? (Already have CVC, Maxima, Minisat, PARI/GP, Prover9.) Bernstein polynomials?
- More advanced decision procedures for new domains like vector spaces.
- Other valuable automated tools like WLOG tactics.

Library improvements

HOL Light already has quite a good library of mathematics, but:

- So much more still to do in advanced geometry (Flyspeck)
- More general measure theory (for probability etc.)
- Some serious algebra, algebraic geometry, topology, . . .
- Public versions of word and floating-point theories

Correctness / proof transfer improvements

- Better HOL-in-HOL proof
- More secure ‘booth mode’ OCaml / HOL-Zero type improvements
- More serious use of proof transfer
- Extend proof transfer to other systems (Coq, Mizar?)