Lem: A Lightweight Tool for Heavyweight Semantics

Scott Owens[1] Peter Böhm[1]
Francesco Zappa Nardelli[2] Peter Sewell[1]

[1]University of Cambridge [2]INRIA

`http://www.cl.cam.ac.uk/~so294/lem/`

23 August, 2011
ITP

Heavyweight Semantics

At Cambridge (with INRIA, NICTA, IBM, and others): TCP; an optical network switch; Java module system; $\text{OCaml}_{\text{light}}$; C/C++ concurrency; x86, POWER and ARM multicore relaxed memory; CompCertTSO verified compiler; C1x semantics

- realistic systems
- large definitions (thousands of lines)
- testing tools/automation
- sometimes proof (later)
- experience with Coq, HOL4, Isabelle/HOL, Ott

Using Lem

Lem: a Lightweight Tool

An engineering challenge

- support human-readable source files
  - simple logic, rich “programming language” features
- take the source text seriously
- support execution
- be quick and predictable
  - catch target errors during translation

e.g., 25 kinds of expressions, 11 patterns

Example in Lem

```
let coherent_memory_use actions lk rf mo hb =
    (* CoRR *)
    ( forall ((x,a) IN rf) ((y,b) IN rf).
      ((a,b) IN hb && same_location a b &&
       is_at_atomic_location lk b)
      -->
      ((x = y) || (x,y) IN mo) ) &&
```

Misleading size : $\approx 1\%$

From C++ M. M. (Batty, Owens, Sarkar, Sewell, Weber in POPL11)

Example in Latex

let $coherent\_memory\_use\ actions\ lk\ rf\ mo\ hb =$

(* CoRR *)

$(\forall (x,\ a) \in rf\ (y,\ b) \in rf.$

$((a,\ b)\ \in\ hb\ \wedge$ same_location $a\ b\ \wedge$

is_at_atomic_location $lk\ b)$

$\longrightarrow$

$((x = y)\ \vee\ (x,\ y)\ \in\ mo))\ \wedge$

Example in OCaml

```
let coherent_memory_use actions lk rf mo hb =
    (* CoRR *)
    ( Pset.for_all (fun (x,a) -> Pset.for_all (fun (y,
      (not
      ( Pset.mem (a,b)  hb && (same_location a b &&
       is_at_atomic_location lk b)) ||
      ((x = y) || Pset.mem  (x,y)  mo))) rf) rf ) &&
```

Example in HOL-4

```
coherent_memory_use actions lk rf mo hb =
   (* CoRR *)
   ( ! ((x,a) :: rf) ((y,b) :: rf).
     ((a,b) IN hb /\ same_location a b /\
      is_at_atomic_location lk b)
     ==>
     ((x = y) \/ (x,y) IN mo) ) /\
```

Example in Isabelle/HOL

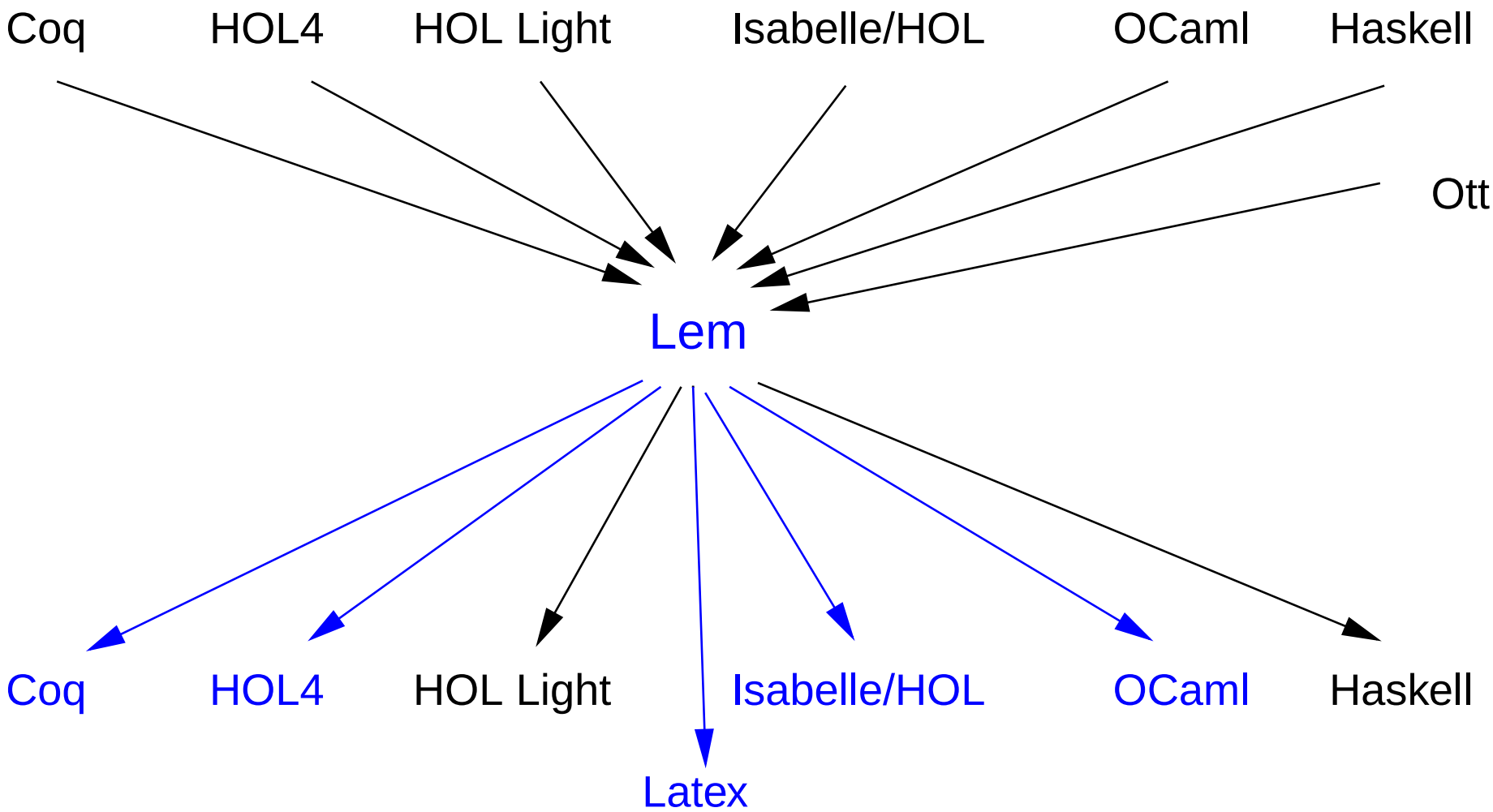
```
definition coherent_memory_use  :: "(Atomic.actio
 " coherent_memory_use actions lk rf mo hb ==
    (* CoRR *)
    (( ALL (x,a) : rf.  ALL (y,b) : rf.
      ( isa_set_mem (a,b)  hb & same_location a b
       is_at_atomic_location lk b)
      -->
      ((x = y) | isa_set_mem  (x,y)  mo)) ) &
```

Features

Logical intersection, syntactic sugar union

- higher-order functions
- recursive functions and inductive relations
- list and set comprehensions
- ML-style polymorphism
- simple type classes
- algebraic and record datatypes

Lem as an Intermediate Language



Lem and Ott

Ott:

- rich syntax support
- lacking sets, functions, polymorphism, etc.
- for PL formalisation only

Lem:

- generally applicable
- rich higher-order logic
- lacking rich syntactic extension

Conclusion

Lem is work in progress

C, C++, POWER, and ARM models

Exports OCaml, HOL-4, Isabelle/HOL, Latex, and soon Coq

Imports . . .

Get Lem alpha version:
`http://www.cl.cam.ac.uk/~so294/lem/`

"Lem"?

Lightweight executable mathematics

Stanisław Lem

Lemma