Metis

System Description

Joe Hurd

Computing Laboratory
Oxford University

ESHOL Workshop
2 December 2005

Talk Plan

1 Example Higher Order Logic Goal

2 Comparison with MESON_TAC

Example Higher Order Logic Goal

Consider the following HOL subgoal:

```
...
1 subgoal:
> val it =
   (!P. (!n. (!m. m < n ==> P m) ==> P n) ==> !n. P n) ==>
    !P. P 0 /\ (!n. P n ==> P (SUC n)) ==> !n. P n
  : goalstack


-  ???
```

Example Higher Order Logic Goal

First, identify relevant lemmas:

```
...
1 subgoal:
> val it =
   (!P. (!n. (!m. m < n ==> P m) ==> P n) ==> !n. P n) ==>
    !P. P 0 /\ (!n. P n ==> P (SUC n)) ==> !n. P n
  : goalstack
```

```
- [LESS_SUC_REFL, num_CASES];
```

```
> val it =
   [|- !n. n < SUC n,
    |- !m. m = 0 \/ ?n. m = SUC n]
   : thm list
```

```
- ???
```

Example Higher Order Logic Goal

Proof 1: The HOL guru way.

```
...
1 subgoal:
> val it =
   (!P. (!n. (!m. m < n ==> P m) ==> P n) ==> !n. P n) ==>
    !P. P 0 /\ (!n. P n ==> P (SUC n)) ==> !n. P n
  : goalstack
```

```
- e (DISCH_THEN (fn t => NTAC 2 STRIP_TAC THEN MP_TAC (Q.ID_SPEC t))
     THEN DISCH_THEN MATCH_MP_TAC
     THEN (Cases THEN1 ASM_REWRITE_TAC [])
     THEN DISCH_THEN (MP_TAC o Q.SPEC `n`)
     THEN ASM_REWRITE_TAC [LESS_SUC_REFL]);
```

```
OK..

Goal proved.
|- (!P. (!n. (!m. m < n ==> P m) ==> P n) ==> !n. P n) ==>
   !P. P 0 /\ (!n. P n ==> P (SUC n)) ==> !n. P n
```

Example Higher Order Logic Goal

Proof 2: A simpler approach.

```
...
1 subgoal:
> val it =
   (!P. (!n. (!m. m < n ==> P m) ==> P n) ==> !n. P n) ==>
    !P. P 0 /\ (!n. P n ==> P (SUC n)) ==> !n. P n
  : goalstack
```

```
- e (METIS_TAC [LESS_SUC_REFL, num_CASES]);
```

```
OK..
metis: m-0-1-2-3-4-5-6r|*|0+7x0+0+0+0+0+0+0+0+0+1+3+1+0+0+
0+3+0+2+2+4+2+0+4+1x2+3+#

Goal proved.
|- (!P. (!n. (!m. m < n ==> P m) ==> P n) ==> !n. P n) ==>
   !P. P 0 /\ (!n. P n ==> P (SUC n)) ==> !n. P n
```

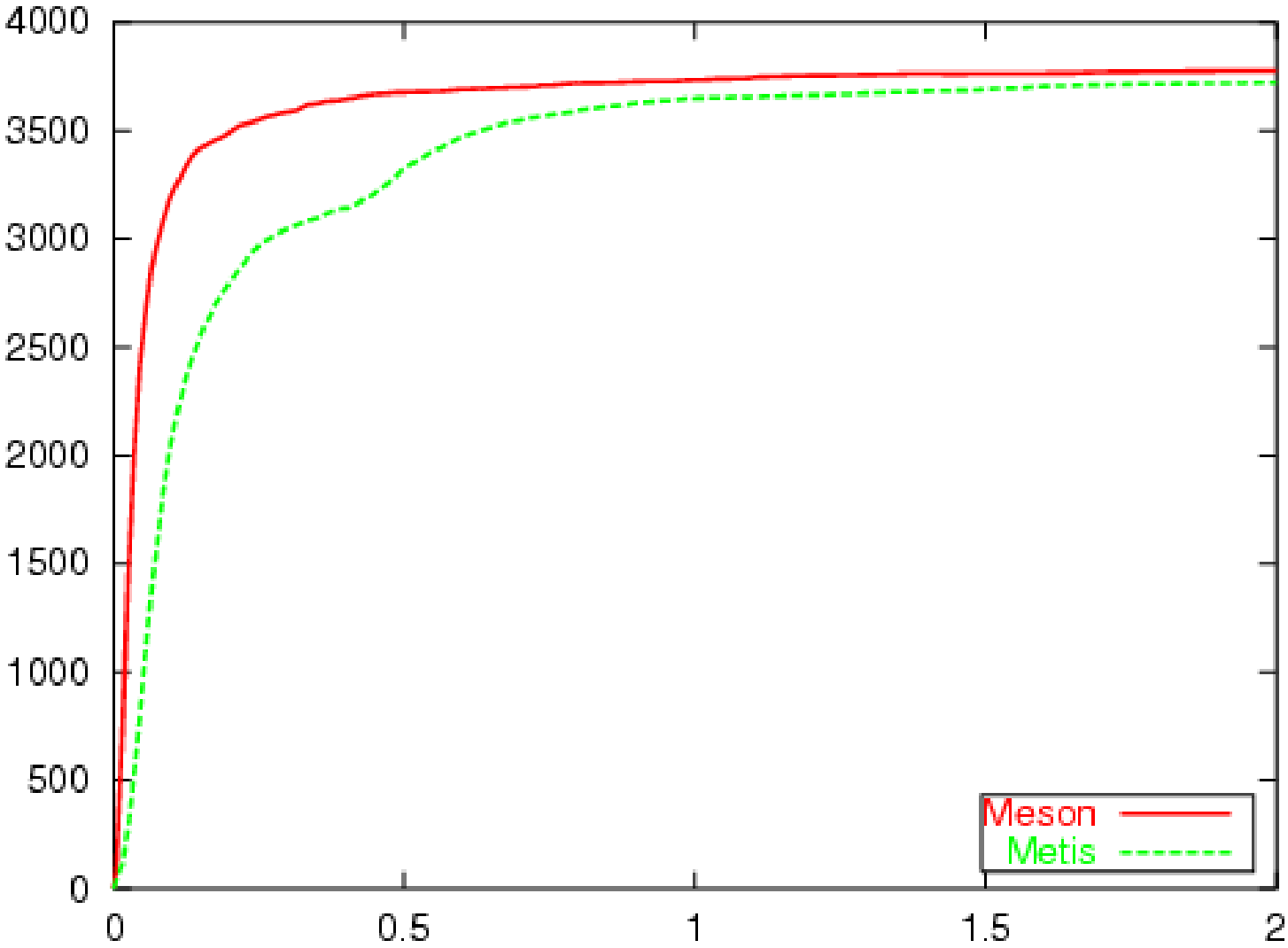
Comparison with MESON_TAC

```
Total subgoals:       1779 + 2024 = 3803
Proved by MESON_TAC: 1779 + 2017 = 3796
Proved by METIS_TAC: 1774 + 2007 = 3781

prob_53(0.02) prob_44(0.02) int_arith_139(0.09)
DeepSyntax_47(0.11) Omega_13(0.11) euclid_8(0.2)
measure_138(0.23) MachineTransition_0(0.29) nc_6(0.38)
prob_169(0.39) prob_170(0.42) fol_1(0.8)
measure_86(0.93) Omega_71(1.78) fol_2(7.63)

TIME DIFFERENCE
Arithmetic mean: 0.30s
Geometric mean:  318%
```

HOL Evaluation



Summary

- Use METIS_TAC in your HOL proofs today!
  - Just do `load "metisLib"; open metisLib;` to make METIS_TAC and METIS_PROVE available.
- No need to retire MESON_TAC.
  - Given the fragile nature of first-order provers, it's quite useful to have two complementary tactics.
- Lots of interesting research in the future work. . .

Future Work

- Specialize first-order prover to create *point tools*:
  - Simple arithmetic reasoning.
  - Support predicate subtyping via always-fire rules.
  - Decision procedure for theories such as finite_map.
- Would really like to store theorems, so the user doesn't have to find the right lemmas each time.
- Improved treatment of combinators at first-order level (pattern unification?).
- Use the interface to create a new link to 'industrial strength' first-order provers.
- More powerful first-order calculus: Knuth-Bendix completion, semantic constraints, etc, etc, . . .