Proof Assistants
for One-and-a-Halfth Order Logic
Work in progress

A. Tasistro

Universidad ORT Uruguay

LSFA 2013
IME - USP
Abstract

I will explain:
- The motivations.
- What One-and-a-Halfth Order Logic is.
- What we have done/are doing.
The context

- A Software Engineering program.
- We insist that they learn *proof* (of programs).
- In particular, *formal* proof.
Two early courses

- Recursion and induction using (concise) Haskell.
- Logic (standard),
  - using Natural Deduction.
  - With a tool for autonomous practice.
Features

- Tree display.
- Backwards/forward derivation.
- Paper and pencil simulation
  - several simultaneous trees,
  - drag and drop.
- Meta-theorems

\[ \forall x (\alpha \land \beta) \vdash \forall x \alpha \land \forall x \beta, \]

not:

\[ \forall x (P(x) \land Q(x)) \vdash \forall x P(x) \land \forall x Q(x), \]

- With re-use.
Technical difficulties with meta-theorems in FOL

∀x(α ⊨ α ∨ ∀xβ) (x not free in α)
∀x(α → β), α[x := t] ⊨ β[x := t].

- Meta-vars (formulæ, terms).
- Side conditions (restricting dependence on variables).
- Explicit substitution.
- Equality, as in:

$$\forall e \frac{\forall x \alpha}{\alpha[x := x]}$$
$$\text{eq} \frac{\alpha[x := t]}{\alpha[x := t]}.$$

- Matching for re-use of meta-theorems.
Lots of didactic proof assistants...

- Pandora
- ADN
- Panda
- Proof Builder
- JAPE
- DC Proof
- Yoda
- ...

but none with the feature.
One-and-a-halfth order logic

- The logic of meta-derivations.
- (We don’t want HOL).
- Gabbay and Mathijssen, 2008.
- Based upon Nominal Terms.
  - atoms,
  - abstraction,
  - swapping,
  - freshness $x \not\# \alpha$,
  - $\alpha$ conversion,
  - unknowns (meta-variables),
  - explicit substitution.

- Judgements $\Gamma \vdash_\Delta \alpha$

\[
\forall i \quad \frac{\Gamma \vdash_\Delta \alpha}{\Gamma \vdash_\Delta \forall x \alpha} \quad \Delta \triangleright x \not\# \Gamma
\]
What have we done?

- (Joint with J. Pais)
  - Full version for propositional logic. (2011, Pais’ Engineering final project.)
  - A prototype for FOL (i.e. One-and-a-halfth —2013, Pais’ Master thesis.)
    - Adapting original system to ND.
    - Implementing matching and equality checking (in presence of explicit substitution).
- We are re-programming the prototype with interface –due to next semester for use in teaching.
- An LCF-like version (embedded in Haskell) for more professional use.
Thanks !!!

tasistro@ort.edu.uy