Using de Bruijn indices

Jérôme Vouillon
CNRS & Université Paris 7
Solution to challenges 1a, 2a, 1b, 2b

Transitivity of subtyping and type soundness, with/without records

Coq proof assistant

De Bruijn indices
Issues

- De Bruijn indices
- Comparison with paper proof
De Bruijn indices

Working with de Bruijn indices is tedious, but manageable
Several possible definitions

- Common

\[(\lambda t)[t'/n] = \lambda (t[t'/n + 1])\]
\[n[t/n] = \uparrow^n t\]

- Better

\[(\lambda t)[t'/n] = \lambda (t[\uparrow t'/n + 1])\]
\[n[t/n] = t\]

A simple rule: \textit{shift indices when crossing an abstraction}

A proof assistant helps a lot: cannot misapply rules
Comparison to Paper Proofs

Overall, follow the structure of the paper proofs

   Enough to find a bug and some inaccuracies

Some differences

   • Avoid environment permutations

   • Lemmas regarding environment manipulation,
     well-formedness of types and terms

Transitivity of subtyping

   By induction on size rather than structure
   of terms
Not as polished as for challenges 1a/2a

Somewhat awkward representation

Binders are implicit, ordered from left to right

\[
\text{let } \{a = (x : T); b = (y : T')\} = \ldots \text{ in } x y
\]

encoded as

\[
\text{let } \{a = T; b = T'\} = \ldots \text{ in } 1 0
\]