Using de Bruijn indices

Jérôme Vouillon
CNRS & Université Paris 7

Submission

- 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

De Bruijn indices

Several possible definitions

- Common

$$
\begin{array}{lcl}
(\lambda \mathtt{t})[\mathtt{t}'/\mathtt{n}] & = & \lambda(\mathtt{t}[\mathtt{t}'/\mathtt{n}+1]) \\
\mathtt{n}[\mathtt{t}/\mathtt{n}] & = & \uparrow^{\mathtt{n}} \mathtt{t}
\end{array}
$$

- Better

$$
\begin{array}{lcl}
(\lambda \mathtt{t})[\mathtt{t}'/\mathtt{n}] & = & \lambda(\mathtt{t}[\uparrow \mathtt{t}'/\mathtt{n}+1]) \\
\mathtt{n}[\mathtt{t}/\mathtt{n}] & = & \mathtt{t}
\end{array}
$$

A simple rule: *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

Records

Not as polished as for challenges 1a/2a

Somewhat awkward representation

Binders are implicit, ordered from left to right

$$\texttt{let}\ \{a = (x : T); b = (y : T')\} = \ldots\ \texttt{in}\ x\ y$$

encoded as

$$\texttt{let}\ \{a = T; b = T'\} = \ldots\ \texttt{in}\ 1\ 0$$