Proof of Type Safety for Extended MinML

Amr Sabry

March 26, 2001

1 Definitions

Syntax:

$$
\begin{array}{rcl}
t & ::= & \mathit{int} \mid \mathit{bool} \mid \alpha \mid t \to t \mid \mathit{rec}\ \alpha.t \mid t\ \mathit{ref} \\
\\
e & ::= & x \mid n \mid o(e_1, \ldots, e_n) \mid \mathit{true} \mid \mathit{false} \mid \mathit{if}\ e\ e\ e \mid \\
 & & \lambda x^t : t.e \mid ee \mid \\
 & & \mathit{roll}\ e \mid \mathit{unroll}\ e \mid \\
 & & \ell \mid \mathit{ref}\ e \mid\ !e \mid e{:=}e
\end{array}
$$

Typing judgments for expressions are of the form $\Lambda; \Gamma \vdash e : t$
Typing judgment for memory is of the form $\vdash M : \Lambda$.
Evaluation judgments are of the form $(M, e) \longmapsto (M', e')$

2 Preservation

Lemma 2.1 (Preservation) *If $\Lambda; \bullet \vdash e : t$ and $\vdash M : \Lambda$ and $(M, e) \longmapsto (M', e')$ then there exists a $\Lambda' \supseteq \Lambda$ such that $\vdash M' : \Lambda'$ and $\Lambda'; \bullet \vdash e' : t$.*

Proof. The proof is by induction on the evaluation judgments. We present one case only:

- Case $(M, +(e_1, e_2)) \longmapsto (M', +(e_1', e_2))$ because $(M, e_1) \longmapsto (M', e_1')$. By assumption:
  - $\Lambda; \bullet \vdash +(e_1, e_2) : t$: By inversion t must be *int* and $\Lambda; \bullet \vdash e_1 : \mathit{int}$ and $\Lambda; \bullet \vdash e_2 : \mathit{int}$.
  - $\vdash M : \Lambda$

  The evaluation judgment $(M, e_1) \longmapsto (M', e_1')$ is shorter, and all the assumptions in the statement of the lemma are satisfied, hence we can apply the inductive hypothesis to conclude that there exists a $\Lambda' \supseteq \Lambda$ such that $\vdash M' : \Lambda'$ and $\Lambda'; \bullet \vdash e_1' : \mathit{int}$. To finish this case, we need to conclude that $\Lambda'; \bullet \vdash +(e_1', e_2) : \mathit{int}$, but this is immediate.