AnML: MinML + Assignments

Andrew W. Appel

October 18, 1999

1 Static Semantics

The typing rules for AnML are the same as for MinML, but with one additional rule:

$$\frac{\Gamma \vdash e_1 : \Gamma(x) \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash (x := e_1;\ e_2) : \tau}\ [\textbf{AsgnTyp}]$$

2 Small-step Dynamic Semantics

$$\frac{}{\Delta, o(v_1, \ldots, v_n) \mapsto \Delta, v}\ [\textbf{OpVals}]$$

(where v is the result of applying o to v_1, …, v_n)

$$\frac{}{\Delta, \texttt{if}\,\texttt{true}\,\texttt{then}\,e_1\,\texttt{else}\,e_2\,\texttt{fi} \mapsto \Delta, e_1}\ [\textbf{IfTrue}]$$

$$\frac{}{\Delta, \texttt{if}\,\texttt{false}\,\texttt{then}\,e_1\,\texttt{else}\,e_2\,\texttt{fi} \mapsto \Delta, e_2}\ [\textbf{IfFalse}]$$

$$\frac{x \notin \mathrm{dom}(\Delta) \quad v = \texttt{fun}\,f\ \texttt{(}x\,\texttt{:}\,\tau_1\texttt{)}\,\texttt{:}\,\tau_2\,\texttt{is}\,e\,\texttt{end}}{\Delta, \texttt{apply(}v, v_1\texttt{)} \mapsto \Delta[x : v_1], [v/f]e}\ [\textbf{CallFun}]$$

Assignment:

$$\frac{\Delta = \Delta'[x : v'] \quad x \notin \mathrm{dom}(\Delta')}{\Delta, (x := v;\ e) \mapsto \Delta'[x : v], e_2}\ [\textbf{AsgnArg}]$$

Search rules:

$$\frac{\Delta, e_i \mapsto \Delta', e_i'}{\Delta, o(v_1, \ldots, v_{i-1}, e_i, \ldots, e_n) \mapsto \Delta', o(v_1, \ldots, v_{i-1}, e_i', \ldots, e_n)}\ [\textbf{OpArg}_i]$$

$$\frac{\Delta, e \mapsto \Delta', e'}{\Delta, \texttt{if}\,e\,\texttt{then}\,e_1\,\texttt{else}\,e_2\,\texttt{fi} \mapsto \Delta', \texttt{if}\,e'\,\texttt{then}\,e_1\,\texttt{else}\,e_2\,\texttt{fi}}\ [\textbf{IfTest}]$$

$$\frac{\Delta, e_1 \mapsto \Delta', e_1'}{\Delta, \texttt{apply(}e_1, e_2\texttt{)} \mapsto \Delta', \texttt{apply(}e_1', e_2\texttt{)}}\ [\textbf{AppFun}]$$

$$\frac{\Delta, e_2 \mapsto \Delta', e_2'}{\Delta, \texttt{apply(}v_1, e_2\texttt{)} \mapsto \Delta', \texttt{apply(}v_1, e_2'\texttt{)}}\ [\textbf{AppArg}]$$

$$\frac{\Delta, e_1 \mapsto \Delta', e_2}{\Delta, (x := e_1;\ e_2) \mapsto \Delta', (x := e_1';\ e_2)}\ [\textbf{AsgnArg}]$$

3 Evaluation Semantics

$$\frac{}{\Delta, x \Downarrow \Delta,\ \Delta(x)}\ [\textbf{EvalVar}] \qquad \frac{}{\Delta, n \Downarrow \Delta, n}\ [\textbf{EvalN}]$$

$$\frac{}{\Delta, \texttt{true} \Downarrow \Delta, \texttt{true}}\ [\textbf{EvalT}] \qquad \frac{}{\Delta, \texttt{false} \Downarrow \Delta, \texttt{false}}\ [\textbf{EvalF}]$$

$$\frac{}{\Delta, \texttt{fun}\, f\ \texttt{(}x\texttt{:}\tau_1\texttt{)}\texttt{:}\tau_2\ \texttt{is}\ e\ \texttt{end} \Downarrow \Delta, \texttt{fun}\, f\ \texttt{(}x\texttt{:}\tau_1\texttt{)}\texttt{:}\tau_2\ \texttt{is}\ e\ \texttt{end}}\ [\textbf{EvalF}]$$

$$\frac{\Delta_0, e_1 \Downarrow \Delta_1, v_1 \quad \ldots \quad \Delta_{n-1}, e_n \Downarrow \Delta_n, v_n}{\Delta_0, o(e_1, \ldots, e_n) \Downarrow \Delta_n, v}\ [\textbf{EvalPrim}]$$

(where v is the result of applying o to $v_1, \ldots, v_n$)

$$\frac{\Delta, e \Downarrow \Delta', \texttt{true} \quad \Delta', e_1 \Downarrow \Delta'', v}{\Delta, \texttt{if}\, e\, \texttt{then}\, e_1\, \texttt{else}\, e_2\, \texttt{fi} \Downarrow \Delta'', v}\ [\textbf{EvalIfTrue}]$$

$$\frac{\Delta, e \Downarrow \Delta', \texttt{false} \quad \Delta', e_2 \Downarrow \Delta'', v}{\Delta, \texttt{if}\, e\, \texttt{then}\, e_1\, \texttt{else}\, e_2\, \texttt{fi} \Downarrow \Delta'', v}\ [\textbf{EvalIfFalse}]$$

$$\frac{\begin{array}{c}\Delta_1, e_1 \Downarrow \Delta_2, v_1 \quad v_1 = \texttt{fun}\, f\ \texttt{(}x\texttt{:}\tau_1\texttt{)}\texttt{:}\tau_2\ \texttt{is}\ e\ \texttt{end} \\ \Delta_2, e_2 \Downarrow \Delta_3, v_2 \quad x \notin \mathrm{dom}(\Delta_2) \quad \Delta_3[x : v_2], [v_1/f]e \Downarrow \Delta_4, v\end{array}}{\Delta_1, \texttt{apply(}e_1, e_2\texttt{)} \Downarrow \Delta_4, v}\ [\textbf{EvalApp}]$$

$$\frac{\Delta, e_1 \Downarrow \Delta_1, v_1 \quad \Delta_1 = \Delta_1'[x : v_1] \quad x \notin \mathrm{dom}(\Delta_1') \quad \Delta_1'[x : v_1], e_2 \Downarrow \Delta', v}{\Delta, (x := e_1;\ e_2) \Downarrow \Delta', v}\ [\textbf{EvalAsgn}]$$