1 Syntax

$$
\begin{aligned}
\text{Program} &::= \{Mod;\}^{*}\ e \\
\text{Module } Mod &::= \text{module } ModName \\
&\qquad \text{interface } [\Delta] \\
&\qquad \text{body } [d] \\
\\
\text{Declaration } \Delta &::= (varident : \tau) \\
&\quad\ \mid (varident : \tau), \Delta \\
\\
\text{Definition } d &::= (varident = e : \tau) \\
&\quad\ \mid (varident = e : \tau), d \\
\\
\text{Expression } e &::= num\ n \mid false \mid true \\
&\quad\ \mid varident \\
&\quad\ \mid ModName.varident \\
&\quad\ \mid this.varident \\
&\quad\ \mid e_1 e_2 \\
&\quad\ \mid (e_1, e_2) \\
&\quad\ \mid \lambda(p)\ .\ e \\
&\quad\ \mid let\ p\ =\ e_1\ in\ e_2 \\
&\quad\ \mid letrec\ p\ =\ e_1\ in\ e_2 \\
&\quad\ \mid if(e_1)\ then\ e_2\ else\ e_3 \\
\\
\text{Pattern } p &::= varident \\
&\quad\ \mid (p, p)
\end{aligned}
$$

2 Type System

$$
\begin{aligned}
\text{Type } \tau \ &::= nat \\
&\mid bool \\
&\mid \tau_1 \to \tau_2 \\
&\mid \tau_1 \times \tau_2 \\
&\mid \alpha \\
\\
\text{Type-Scheme } \sigma \ &::= \tau \\
&\mid \forall \alpha.\sigma \\
\\
\text{Context } \Gamma &::= \cdot \\
&\mid \Gamma : (x : \tau) \\
\\
\text{Typing } &::= \Gamma \vdash e : \sigma \\
\\
\text{ModuleTyping } &::= \Gamma \gg Mod \\
\text{DefinitionTyping } &::= \Gamma \models d \to \Gamma' \\
\\
\text{DeclarationTyping } &::= \Gamma \propto \Delta
\end{aligned}
$$

2.1 Rules

$$\Gamma \vdash true : bool \tag{T-True}$$

$$\Gamma \vdash false : bool \tag{T-False}$$

$$\Gamma \vdash num\ n : nat \tag{T-Num}$$

$$\frac{\sigma \geq \tau \qquad varident : \sigma \in \Gamma}{\Gamma \vdash varident : \tau} \tag{T-Mono}$$

$$\frac{\Gamma \vdash e_1 : \tau_2 \to \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash e_1 e_2 : \tau_1} \tag{T-App}$$

$$varident : \sigma \to \cdot : (varident : \sigma) \tag{T-BuildContext1}$$

$$\frac{p_1 : \sigma_1 \to \Gamma_1 \qquad p_2 : \sigma_2 \to \Gamma_2 \qquad \Gamma_3 = \Gamma_1 \bigcup \Gamma_2}{(p_1, p_2) : \sigma_1 \times \sigma_2 \to \Gamma_3} \tag{T-BuildContext2}$$

$$\frac{p : \tau_2 \to \Gamma_2 \qquad \Gamma_2 \bigcup \Gamma_1 \vdash e : \tau_1}{\Gamma_1 \vdash \lambda(p).e : \tau_2 \to \tau_1} \tag{T-Fun}$$

$$\frac{\Gamma \vdash e_1 : bool \qquad \Gamma \vdash e_2 : \tau \qquad \Gamma \vdash e_3 : \tau}{\Gamma \vdash if\ e_1\ then\ e_2\ else\ e_3 : \tau} \tag{T-IfThenElse}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2} \tag{T-Pair}$$

$$\frac{\Gamma \vdash e_2 : \tau_2 \quad \sigma = gen(\Gamma, \tau) \quad p : \sigma \to \Gamma_2 \quad \Gamma \bigcup \Gamma_2 \vdash e_1 : \tau}{\Gamma \vdash let\ p\ =\ e_2\ in\ e_1 : \tau} \tag{T-Let}$$

$$\frac{\Gamma \vdash let\ p\ =\ fix\ (\lambda p.e_2)\ in\ e_1 : \tau}{\Gamma \vdash letrec\ p\ =\ e_2\ in\ e_1 : \tau} \tag{T-Letrec}$$

$$\frac{\Gamma \vdash e : \tau \to \tau}{\Gamma \vdash fix(e) : \tau} \tag{T-Fix}$$

$$\frac{\sigma \geq \tau \qquad this.varident : \sigma \in \Gamma}{\Gamma \vdash this.varident : \tau} \qquad \text{(T-ModVarThis)}$$

$$\frac{\Gamma \gg ModName \qquad varident : \tau \in ModName.interface}{\Gamma \vdash ModName.varident : \tau} \qquad \text{(T-ModVarOther)}$$

$$\frac{\cdot \models ModName.body \to \Gamma' \qquad \Gamma' \propto ModName.interface}{\Gamma \gg ModName} \qquad \text{(T-Module)}$$

$$\frac{(x : \tau) \in \Gamma}{\Gamma \propto (x : \tau), \Delta} \qquad \text{(T-ModInterface)}$$

$$\frac{(x : \tau) \in \Gamma}{\Gamma \propto (x : \tau)} \qquad \text{(T-ModInterfaceSingle)}$$

$$\frac{\Gamma' = \Gamma : (x : \tau) \qquad \Gamma' \models d \qquad \Gamma \vdash e : \tau}{\Gamma \models (x = e : \tau), d \to \Gamma'} \qquad \text{(T-ModBody)}$$

$$\frac{\Gamma' = \Gamma : (x : \tau) \qquad \Gamma \vdash e : \tau}{\Gamma \models (x = e : \tau) \to \Gamma'} \qquad \text{(T-ModBodySingle)}$$

3 Operational semantics

$$\begin{aligned} \text{Value } v ::= &\ numn \mid true \mid false \\ &|(v, v) \\ &|\lambda p.e \end{aligned}$$

$$\begin{aligned} \text{Environment } E \ ::= &\ \cdot \\ &\mid E : (varident = v) \\ &\mid E : (this.varident = v) \end{aligned}$$

$$\text{Evaluation } ::= e \to e'$$

3.1 Rules

$$if\ true\ then\ e_1\ else\ e_2 \rightarrow e_1 \tag{E-IfTrue}$$

$$if\ false\ then\ e_1\ else\ e_2 \rightarrow e_2 \tag{E-IfFalse}$$

$$\frac{e_1 \rightarrow e_1'}{if\ e_1\ then\ e_2\ else\ e_3 \rightarrow if\ e_1'\ then\ e_2\ else\ e_3} \tag{E-IfThenElse}$$

$$\frac{e_1 \rightarrow e_1'}{(e_1, e_2) \rightarrow (e_1', e_2)} \tag{E-PairLeft}$$

$$\frac{e_2 \rightarrow e_2'}{(e_1, e_2) \rightarrow (e_1, e_2')} \tag{E-PairRight}$$

$$\frac{e_1 \rightarrow e_1'}{let\ p\ =\ e_1\ in\ e_2 \rightarrow let\ x\ =\ e_1'\ in\ e_2} \tag{E-Let}$$

$$let\ x\ =\ v\ in\ e \rightarrow [x \mapsto v]e \tag{E-LetV}$$

$$letrec\ p =\ e_1\ in\ e_2 \rightarrow let\ p\ = fix(\lambda p.e_1)\ in\ e_2 \tag{E-LetRec}$$

$$\frac{e \rightarrow e'}{fix(e) \rightarrow fix(e')} \tag{E-Fix}$$

$$fix(\lambda(p.e)) \rightarrow [p \mapsto (fix(\lambda(p.e))]e \tag{E-FixRec}$$

$$let\ (p_1, p_2)\ =\ (e_1, e_2)\ in\ e_3 \rightarrow let\ p_1\ =\ e_1\ in\ (let\ p_2\ =\ e_2\ in\ e_3) \tag{E-PatternMatch}$$

$$\frac{e_1 \rightarrow e_1'}{e_1 e_2 \rightarrow e_1' e_2} \tag{E-App1}$$

$$\frac{e_2 \rightarrow e_2'}{v\ e_2 \rightarrow v\ e_2'} \tag{E-App2}$$

$$(\lambda x.e)\ v \rightarrow [x \mapsto v]e \tag{E-Lambda}$$

$$(\lambda(p_1, p_2).e_3)\ (e_1, e_2) \rightarrow (\lambda p_1.(\lambda p_2.e_3)\ e_2)\ e_1 \tag{E-MatchLambda}$$

$$\frac{(x = e' : \tau) \in ModName.body \qquad e = [this.y \mapsto M.y]e' \ \forall(this.y \in e')}{M.x \rightarrow e} \tag{E-ModVar}$$