

Sémantique des langages de programmation

1 Syntaxe du langage mini-ML

1.1 Langage complet

La syntaxe du langage mini-ML que nous utiliserons comme support est définie par la grammaire suivante :

$$\begin{aligned} Expr &\rightarrow Ident \\ &| Const \\ &| Expr \text{ Binaire } Expr \\ &| Unaire Expr \\ &| \text{fun } Ident \rightarrow Expr \\ &| (Expr) Expr \\ &| \text{if } Expr \text{ then } Expr \text{ else } Expr \\ &| \text{let } Ident = Expr \text{ in } Expr \\ &| \text{letrec } Ident = Expr \text{ in } Expr \\ &| (Expr) \\ &| Expr ; Expr \\ &| \text{ref } Expr \\ &| ! Expr \\ &| Ident := Expr \\ \\ Const &\rightarrow entier \mid boolean \\ \\ Unaire &\rightarrow - \mid ! \\ \\ Binaire &\rightarrow + \mid - \mid * \mid / \mid \% \mid \& \mid | \\ &| == \mid != \mid < \mid <= \mid > \mid >= \end{aligned}$$

Important : Les quatre dernières constructions de la syntaxe de mini-ML correspondent à des effets de bord, elles ne seront pas traitées dans toutes les sémantiques.

1.2 Valeurs

1.2.1 Avec effets de bord

Lorsque des effets de bord peuvent se produire, il faut compléter le programme en cours d'évaluation par une représentation du contexte d'évaluation qui peut être modifié. Nous considérons ici la mémoire.

Les valeurs possibles pour une expression en langage mini-ML avec effets de bord sont décrites par la syntaxe suivante :

$$\begin{aligned} Valeur &\rightarrow [Const, Mem] \\ &| [\langle \text{fun } Ident \rightarrow Expr, Env \rangle, Mem] \\ &| [\perp, Mem] \end{aligned}$$

2 Sémantique opérationnelle de mini-ML avec effets de bord

2.1 Formalisme de déduction

Un jugement d'évaluation s'écrit sous la forme $\gamma \vdash [e, m_1] \Rightarrow [v, m_2]$.

Rappelons l'ensemble des axiomes et des règles de déduction correspondant à l'évaluation de mini-ML avec effets de bord.

$$\frac{x \in \gamma_1 \quad \gamma_1(x) = \langle e, \gamma_2 \rangle \quad \gamma_2 \vdash [e, m] \Rightarrow [v, m]}{\gamma_1 \vdash [x, m] \Rightarrow [v, m]}$$

$$\frac{x \in \gamma_1 \quad \gamma_1(x) = v \quad v \neq \langle e, \gamma_2 \rangle}{\gamma_1 \vdash [x, m] \Rightarrow [v, m]}$$

$$\frac{x \notin \gamma}{\gamma \vdash [x, m] \Rightarrow [\perp, m]}$$

$$\frac{\gamma \vdash [e_1, m_1] \Rightarrow [v_1, m_2] \quad \gamma \vdash [e_2, m_2] \Rightarrow [v_2, m_3] \quad v_1 \times v_2 \in \text{dom } op \quad v = v_1 \text{ op } v_2}{\gamma \vdash [e_1 \text{ op } e_2, m_1] \Rightarrow [v, m_3]}$$

$$\frac{\gamma \vdash [e, m] \Rightarrow [v, m'] \quad v \in \text{dom } op \quad v' = op \, v}{\gamma \vdash [op \, e, m] \Rightarrow [v', m']}$$

$$\gamma \vdash [\text{fun } x \rightarrow e, m] \Rightarrow [\langle \text{fun } x \rightarrow e, \gamma \rangle, m]$$

$$\frac{\gamma_1 \vdash [e_1, m_1] \Rightarrow [\langle \text{fun } x \rightarrow e_3, \gamma_2 \rangle, m_2] \quad \gamma_1 \vdash [e_2, m_2] \Rightarrow [v, m_3] \quad \gamma_2 :: \{x \mapsto v\} \vdash [e_3, m_3] \Rightarrow [e_4, m_4]}{\gamma_1 \vdash [(e_1 \rightarrow e_2), m] \Rightarrow [e_4, m_4]}$$

$$\frac{\gamma \vdash [e_1, m_1] \Rightarrow [v, m_2] \quad \gamma :: \{x \mapsto v\} \vdash [e_2, m_2] \Rightarrow [e_3, m_3]}{\gamma \vdash [\text{let } x = e_1 \text{ in } e_2, m_1] \Rightarrow [e_3, m_3]}$$

$$\frac{\gamma :: \{x \mapsto \langle \text{letrec } x = e_1 \text{ in } e_1, \gamma \rangle\} \vdash [e_2, m_1] \Rightarrow [e_3, m_2]}{\gamma \vdash [\text{letrec } x = e_1 \text{ in } e_2, m_1] \Rightarrow [e_3, m_2]}$$

$$\frac{\gamma \vdash [e_1, m_1] \Rightarrow [\text{true}, m_2] \quad \gamma \vdash [e_2, m_2] \Rightarrow [e_4, m_3]}{\gamma \vdash [\text{if } e_1 \text{ then } e_2 \text{ else } e_3, m_1] \Rightarrow [e_4, m_3]} \quad \frac{\gamma \vdash [e_1, m_1] \Rightarrow [\text{false}, m_2] \quad \gamma \vdash [e_3, m_2] \Rightarrow [e_4, m_3]}{\gamma \vdash [\text{if } e_1 \text{ then } e_2 \text{ else } e_3, m_1] \Rightarrow [e_4, m_3]}$$

$$\frac{\gamma \vdash [e, m_1] \Rightarrow [\text{! } r, m_2] \quad r \in m_2 \quad v = m_2(r)}{\gamma \vdash [\text{! } e, m_1] \Rightarrow [v, m_2]}$$

$$\frac{\gamma \vdash [e_1, m_1] \Rightarrow [\text{! } r, m_2] \quad \gamma \vdash [e_2, m_2] \Rightarrow [v, m_3]}{\gamma \vdash [e_1 := e_2, m_1] \Rightarrow [(), m_3 :: \{r \mapsto v\}]}$$

$$\frac{\gamma \vdash [e_1, m_1] \Rightarrow [(), m_2] \quad \gamma \vdash [e_2, m_2] \Rightarrow [v, m_3]}{\gamma \vdash [e_1 ; e_2, m_1] \Rightarrow [v, m_3]}$$

$$\frac{\gamma \vdash [e, m_1] \Rightarrow [v, m_2] \quad r \notin m_2}{\gamma \vdash [\text{ref } e, m_1] \Rightarrow [\text{! } r, m_2 :: \{r \mapsto v\}]}$$

3 Typage de mini-ML

3.1 Forme des types

Les types possibles pour une expression du langage mini-ML sont décrits par la syntaxe suivante :

$$\begin{array}{lcl} \tau & \rightarrow & \alpha \mid \text{int} \mid \text{bool} \\ & & \mid \tau \rightarrow \tau \\ & & \mid (\tau) \end{array}$$

3.2 Sémantique axiomatique du typage

Un jugement de typage s'écrit sous la forme $\sigma \vdash e : \tau$.

Rappelons l'ensemble des axiomes et des règles de déduction correspondant au typage de mini-ML.

$$\frac{x \in \sigma \quad \sigma(x) = \tau}{\sigma \vdash x : \tau}$$

$$\frac{\sigma \vdash e_1 : \tau_1 \quad \sigma \vdash e_2 : \tau_2 \quad \tau_1 \times \tau_2 = \text{dom } op \quad \tau = \text{codom } op}{\sigma \vdash e_1 \text{ } op \text{ } e_2 : \tau}$$

$$\frac{\sigma \vdash e : \tau \quad \tau = \text{dom } op \quad \tau' = \text{codom } op}{\sigma \vdash op \text{ } e : \tau'}$$

$$\frac{\sigma :: \{x : \tau_1\} \vdash e : \tau_2}{\sigma \vdash \mathbf{fun } x \rightarrow e : \tau_1 \rightarrow \tau_2} \quad \frac{\sigma \vdash e_1 : \tau_1 \rightarrow \tau_2 \quad \sigma \vdash e_2 : \tau_1}{\sigma \vdash (e_1 \text{ }) e_2 : \tau_2}$$

$$\frac{\sigma \vdash e_1 : \tau_1 \quad \sigma :: \{x : \tau_1\} \vdash e_2 : \tau_2}{\sigma \vdash \mathbf{let } x = e_1 \text{ in } e_2 : \tau_2}$$

$$\frac{\sigma :: \{x : \tau_1\} \vdash e_1 : \tau_1 \quad \sigma :: \{x : \tau_1\} \vdash e_2 : \tau_2}{\sigma \vdash \mathbf{letrec } x = e_1 \text{ in } e_2 : \tau_2}$$

$$\frac{\sigma \vdash e_1 : \mathbf{bool} \quad \sigma \vdash e_2 : \tau \quad \sigma \vdash e_3 : \tau}{\sigma \vdash \mathbf{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau}$$