Sémantique et traduction des langages
Travaux dirigés
Interprétation avec effets de bord et typage

Objectif : L'objectif principal de ce thème est d'apprendre à exprimer la sémantique d'exécution en présence d'effets de bord(sémantique dynamique) et de vérification par typage (sémantique statique) d'un langage en exploitant la notation de la déduction naturelle.

Le langage considéré est un sous-ensemble de ML étudié en cours que nous allons enrichir de manière progressive.

Nous reprendrons également le codage en langage ML de cette sémantique qui sera ensuite mis en oeuvre en TP.

1 Rappel : Le langage mini-ML

1.1 Syntaxe du langage

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

$$
\begin{array}{lcl}
\textit{Expr} & \rightarrow & \textit{Ident} \\
 & | & \textit{Const} \\
 & | & \textit{Expr Binaire Expr} \\
 & | & \textit{Unaire Expr} \\
 & | & \texttt{(}\ \textit{Expr}\ \texttt{)} \\
 & | & \texttt{if}\ \textit{Expr}\ \texttt{then}\ \textit{Expr}\ \texttt{else}\ \textit{Expr} \\
 & | & \texttt{let}\ \textit{Ident}\ \texttt{=}\ \textit{Expr}\ \texttt{in}\ \textit{Expr} \\
 & | & \texttt{fun}\ \textit{Ident}\ \texttt{->}\ \textit{Expr} \\
 & | & \texttt{(}\ \textit{Expr}\ \texttt{)}\ \textit{Expr} \\
 & | & \texttt{let rec}\ \textit{Ident}\ \texttt{=}\ \textit{Expr}\ \texttt{in}\ \textit{Expr} \\
 & & \\
\textit{Const} & \rightarrow & \textit{entier} \mid \textit{booleen} \\
 & & \\
\textit{Unaire} & \rightarrow & \texttt{-} \mid \texttt{!} \\
 & & \\
\textit{Binaire} & \rightarrow & \texttt{+} \mid \texttt{-} \mid \texttt{*} \mid \texttt{/} \mid \texttt{\%} \mid \texttt{\&} \mid \texttt{|} \\
 & | & \texttt{==} \mid \texttt{!=} \mid \texttt{<} \mid \texttt{<=} \mid \texttt{>} \mid \texttt{>=}
\end{array}
$$

1.2 Représentation des valeurs

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

$$
\begin{array}{lcl}
\textit{Valeur} & \rightarrow & \textit{Const} \\
 & | & \langle\, \textit{Expr}\ ,\ \textit{Env}\, \rangle \\
 & | & \bot
\end{array}
$$

2 Sémantique opérationnelle

Un jugement d'évaluation s'écrit sous la forme $\gamma \vdash e \Downarrow v$.

2.1 Rappel

Rappelons l'ensemble des axiomes et des règles de déduction correspondant à l'évaluation de mini-ML présentée en cours.

$$\gamma \vdash \mathit{entier} \Downarrow \mathit{entier} \qquad \gamma \vdash \mathit{booleen} \Downarrow \mathit{booleen}$$

$$\frac{x \in \gamma \quad \gamma(x) = \langle\, e\,,\, \gamma_{def} \,\rangle \quad \gamma_{def} \vdash e \Downarrow v}{\gamma \vdash x \Downarrow v} \qquad \frac{x \in \gamma \quad \gamma(x) = v \quad v \neq \langle\, e\,,\, \gamma_{def} \,\rangle}{\gamma \vdash x \Downarrow v}$$

$$\frac{\gamma \vdash e_2 \Downarrow v_2 \quad v_2 \neq \bot \qquad \gamma \vdash e_1 \Downarrow v_1 \quad v_1 \neq \bot \qquad v_1 \times v_2 \in \mathit{dom}\, op \quad v = v_1\, op\, v_2}{\gamma \vdash e_1\ op\ e_2 \Downarrow v}$$

$$\frac{\gamma \vdash e \Downarrow v \quad v \neq \bot \quad v \in \mathit{dom}\, op \quad v' = op\, v}{\gamma \vdash op\ e \Downarrow v'}$$

$$\frac{\gamma \vdash e_1 \Downarrow \texttt{true} \quad \gamma \vdash e_2 \Downarrow v}{\gamma \vdash \texttt{if}\ e_1\ \texttt{then}\ e_2\ \texttt{else}\ e_3 \Downarrow v} \qquad \frac{\gamma \vdash e_1 \Downarrow \texttt{false} \quad \gamma \vdash e_3 \Downarrow v}{\gamma \vdash \texttt{if}\ e_1\ \texttt{then}\ e_2\ \texttt{else}\ e_3 \Downarrow v}$$

$$\frac{\gamma \vdash e_1 \Downarrow v_1 \quad v_1 \neq \bot \quad \gamma :: \{x \mapsto v_1\} \vdash e_2 \Downarrow v}{\gamma \vdash \texttt{let}\ x\ \texttt{=}\ e_1\ \texttt{in}\ e_2 \Downarrow v}$$

$$\gamma \vdash \texttt{fun}\ x\ \texttt{->}\ e \Downarrow \langle\, \texttt{fun}\ x\ \texttt{->}\ e\,,\, \gamma \,\rangle$$

$$\frac{\gamma \vdash e_2 \Downarrow v_2 \quad v_2 \neq \bot \quad \gamma \vdash e_1 \Downarrow \langle\, \texttt{fun}\ x\ \texttt{->}\ e_3\,,\, \gamma_{def} \,\rangle \quad \gamma_{def} :: \{x \mapsto v_2\} \vdash e_3 \Downarrow v}{\gamma \vdash \texttt{(}\ e_1\ \texttt{)}\ e_2 \Downarrow v}$$

$$\frac{\gamma :: \{x \mapsto \langle\, \texttt{let rec}\ x\ \texttt{=}\ e_1\ \texttt{in}\ e_1\,,\, \gamma \,\rangle\} \vdash e_2 \Downarrow v}{\gamma \vdash \texttt{let rec}\ x\ \texttt{=}\ e_1\ \texttt{in}\ e_2 \Downarrow v}$$

Il faut ajouter à ces règles, celles d'apparition et propagation des erreurs.

2.2 Ajout des effets de bord

$$\begin{array}{rcl} \mathit{Expr} & \rightarrow & \texttt{ref}\ \mathit{Expr} \\ & | & \texttt{!}\ \mathit{Expr} \\ & | & \mathit{Expr}\ \texttt{:=}\ \mathit{Expr} \\ & | & \mathit{Expr}\ \texttt{;}\ \mathit{Expr} \\ & | & \texttt{while}\ \mathit{Expr}\ \texttt{do}\ \mathit{Expr}\ \texttt{done} \end{array}$$

Proposer des règles d'exécution pour la création de référence sur une valeur, la lecture de la valeur référencée, la modification de la valeur référencée, et le séquencement d'expressions. Vous ne traiterez les cas d'erreurs que pour le séquencement d'expressions.

3 Vérification par typage

3.1 Forme des types

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

$$\begin{array}{rcl} \tau & \rightarrow & \alpha \mid \texttt{int} \mid \texttt{bool} \mid \texttt{unit} \\ & | & \tau \rightarrow \tau \\ & | & (\tau) \end{array}$$

3.2 Règles de 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.

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

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

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

3.3 Exercice : Ajout de la définition simple

Proposer des règles de typage pour la définition simple.

3.4 Exercice : Ajout de la conditionnelle

Proposer des règles de typage pour la conditionnelle.

3.5 Exercice : Ajout de la définition et l'appel de fonction

Proposer des règles de typage pour la définition et l'appel de fonction.

3.6 Exercice : Ajout de la définition récursive

Proposer des règles de typage pour la conditionnelle.

3.7 Exercice : Ajout des effets de bord

Proposer des règles de typage pour l'allocation, la lecture et l'écriture de références ainsi que pour le séquencement d'expressions.