Sémantique et traduction des langages
Travaux dirigés
Sémantique Opérationnelle et Interprétation

Objectif : L'objectif principal de ce thème est d'apprendre à exprimer la sémantique d'exécution (sémantique dynamique) d'un langage en exploitant la notation de la déduction naturelle. Il s'agit d'une sémantique opérationnelle à grand pas qui calcule la valeur de chaque expression.

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 œuvre en TP.

1 Rappel de cours : 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 :

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

1.2 Représentation des valeurs

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

$$
\begin{array}{lcl}
\textit{Valeur} & \rightarrow & \textit{Const} \\
 & | & \bot
\end{array}
$$

Cette syntaxe pourra être étendue si nécessaire.

2 Sémantique opérationnelle

Un jugement d'évaluation s'écrit sous la forme $\gamma \vdash e \Downarrow v$, où :
— γ est un environnement (association $\textit{Ident}$ / $\textit{Valeur}$) ;
— e est une expression ($\textit{Expr}$) ;
— v est une valeur ($\textit{Valeur}$).

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

Pour mémoire, le domaine d'un opérateur, utilisé pour la vérification de type, contient les erreurs $\bot_{runtime}$ qui peuvent se produire lors d'une exécution bien typée. Par exemple, la division par zéro ou les dépassements de capacité en arithmétique bornée, font parti du domaine des opérateurs arithmétiques.

Notons que les règles explicitent dans le traitement des erreurs le fait que le langage mini-ML calcule les paramètres des opérateurs binaires de droite à gauche. Autrement dit, si une erreur apparait lors du calcul du paramètre de droite, le paramètre de gauche ne sera pas calculé.

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

$$\frac{x \in \gamma \quad \gamma(x) = v}{\gamma \vdash x \Downarrow v} \qquad \frac{x \notin \gamma}{\gamma \vdash x \Downarrow \bot_{undef}}$$

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

$$\frac{\gamma \vdash e \Downarrow v \quad v \neq \bot \quad v \notin \mathit{dom}\, op}{\gamma \vdash op\, e \Downarrow \bot_{type}} \qquad \frac{\gamma \vdash e \Downarrow v \quad v = \bot_c}{\gamma \vdash op\, e \Downarrow \bot_c}$$

$$\frac{\begin{array}{cc}\gamma \vdash e_2 \Downarrow v_2 & \gamma \vdash e_1 \Downarrow v_1 \\ v_2 \neq \bot & v_1 \neq \bot\end{array} \quad v_1 \times v_2 \in \mathit{dom}\, op \quad v = v_1\, op\, v_2}{\gamma \vdash e_1\, op\, e_2 \Downarrow v}$$

$$\frac{\gamma \vdash e_1 \Downarrow v_1 \quad \gamma \vdash e_2 \Downarrow v_2 \quad v_1 \neq \bot \quad v_2 \neq \bot \quad v_1 \times v_2 \notin \mathit{dom}\, op}{\gamma \vdash e_1\, op\, e_2 \Downarrow \bot_{type}}$$

$$\frac{\gamma \vdash e_2 \Downarrow v_2 \quad v_2 = \bot_c}{\gamma \vdash e_1\, op\, e_2 \Downarrow \bot_c} \qquad \frac{\begin{array}{c}\gamma \vdash e_2 \Downarrow v_2 \\ v_2 \neq \bot\end{array} \quad \gamma \vdash e_1 \Downarrow v_1 \quad v_1 = \bot_c}{\gamma \vdash e_1\, op\, e_2 \Downarrow \bot_c}$$

2.1 Exemple d'exploitation des règles

Construire la dérivation permettant de calculer la valeur de l'expression $-\ 1\ +\ n\ *\ 2$ dans l'environnement $\{n \mapsto 3\}$.

2.2 Ajout de la définition locale

$$\mathit{Expr} \quad \rightarrow \quad \texttt{let}\ \mathit{Ident}\ \texttt{=}\ \mathit{Expr}\ \texttt{in}\ \mathit{Expr}$$

Vous traiterez également les cas d'erreurs.

1. Quelles sont les principales étapes d'exécution pour une définition locale ?
2. Quelles sont les différentes variantes pour l'exécution d'une définition locale ?
3. Proposer des règles d'exécution pour la définition locale.

2.3 Ajout de la conditionnelle

$$\mathit{Expr} \quad \rightarrow \quad \texttt{if}\ \mathit{Expr}\ \texttt{then}\ \mathit{Expr}\ \texttt{else}\ \mathit{Expr}$$

Vous traiterez également les cas d'erreurs.

1. Quelles sont les principales étapes d'exécution pour une conditionnelle ?
2. Quelles sont les différentes variantes pour l'exécution d'une conditionnelle ?
3. Proposer des règles d'exécution pour la conditionnelle.

2.4 Ajout de la définition et l'appel de fonctions

$$\begin{array}{rcl} \mathit{Expr} & \rightarrow & \texttt{fun}\ \mathit{Ident}\ \texttt{->}\ \mathit{Expr} \\ & | & \texttt{(}\ \mathit{Expr}\ \texttt{)}\ \mathit{Expr} \end{array}$$

Vous traiterez également les cas d'erreurs.

1. Quelles sont les principales étapes d'exécution pour une definition et un appel de fonction ?
2. Quelles sont les différentes variantes pour l'exécution d'une définition et d'un appel de fonction ?
3. Proposer des règles d'exécution pour les définition et appel de fonctions.

2.5 Ajout de la définition récursive

$$\mathit{Expr} \rightarrow \texttt{let rec}\ \mathit{Ident}\ \texttt{=}\ \mathit{Expr}\ \texttt{in}\ \mathit{Expr}$$

Vous traiterez également les cas d'erreurs.

1. Proposer un exemple simple de fonction récursive et détailler un exemple d'exécution pour un appel avec 2 appels récursifs.
2. Quelles sont les principales étapes d'exécution pour une définition récursive ?
3. Quelles sont les différentes variantes pour l'exécution d'une définition récursive ?
4. Proposer des règles d'exécution pour la définition récursive.