

Exercices sémantique

2023-2024

Note : ces exercices (sauf l'exercice 3 – qui est là pour illustrer que le concept qu'on voit ici est général) parlent du langage défini dans le cours (référez-vous au polycopié pour la définition et la sémantique du langage). Les programmes décrits ici sont également présentés dans la base de code du compilateur qu'on implémentera (sous les noms imaginatifs de `exoX_Y` pour être bien repérables). Le but est que vous puissiez comparer les résultats de vos exercices sur papier avec votre implémentation, de manière à ce que les deux s'entre-guident. De même, vous pouvez voir une version lisible des programmes présentés ici dans le dossier `programs` du projet (dans les sous-dossiers `correct` de chaque sous-dossier).

Tout n'est pas à faire en une fois, le planning prévisionnel est d'aborder `LCalc` sur la première semaine, `LArray` et `LBranch` sur la suivante, et enfin `LFonc` sur la troisième.

Exercice 1 : Expressions simples

On considère le langage du cours (référez-vous au polycopié).

Pour simplifier les notations, on va omettre (sur papier) les constructeurs de constantes et de variables. Ainsi, on notera `4` au lieu de `Cst_i(4)`, `true` au lieu de `Cst_b(true)`, et `x` au lieu de `Var(x)`. On supposera qu'aucune variable ne se nomme `true` ou `false`...

On considère l'expression suivante :

```
Binop(Add,Binop(Mul,Unop(UMin,4),4),5)
```

Dessinez l'arbre correspondant. En vous référant au polycopié, (le tableau de sémantique) déterminez la valeur de cette expression (dans un environnement quelconque, puisqu'il n'y a pas de variable) en fonction des opérations de la machine abstraite.

Même question pour :

```
Binop(And,Binop(Eq,Binop(Add,4.2,1.0),5.2),Unop(Not,false))
```

Même question pour :

```
Binop(Or,Binop(Leq,4,5),Binop(Neq,5,5))
```

Exercice 2 : Variables

Toujours dans le langage du cours.

Même questions que précédemment, mais avec l'expression :

$\text{Binop}(\text{Sub}, x, y)$

Dans les environnements suivants :

- $\rho_1 ::= \langle x \rightarrow 4, y \rightarrow 2 \rangle$
- $\rho_2 ::= \langle x \rightarrow 4.2, y \rightarrow 3.4 \rangle$
- $\rho_3 ::= \langle x \rightarrow \text{true}, y \rightarrow 4 \rangle$
- $\rho_4 ::= \langle x \rightarrow 1 \rangle$

Exercice 3 : Expressions exotiques

On considère le mini-langage d'expressions suivant :

$\text{expr} ::= \text{Cst}(n)$ où $n \in \mathbb{Z}$
 $\text{Var}(x)$ où x est un nom
 $\text{A}(\text{expr}, \text{expr})$
 $\text{B}(\text{expr}, \text{expr})$

On considère que ces expressions vont s'évaluer dans \mathbb{Z} . La mémoire est ici un tableau ρ indexé par les noms et à valeur dans \mathbb{Z} .

Voici leur sémantique :

$[\text{Cst}(n)]_\rho ::= n$
 $[\text{Var}(x)]_\rho ::= \rho(x)$
 $[\text{A}(e1, e2)]_\rho ::= 2 * [e1]_\rho + [e2]_\rho$
 $[\text{B}(e1, e2)]_\rho ::= [e1]_\rho - 2 * [e2]_\rho$

où $+$, $*$ et $-$ désignent les opérations usuelles sur \mathbb{Z} .

Donnez, en détaillant les étapes de calcul la valeur de l'expression

$\text{A}(\text{A}(\text{Cst}(3), \text{Var}(x)), \text{B}(\text{Var}(y), \text{B}(\text{Cst}(4), \text{Cst}(3))))$

dans l'environnement $\rho = \langle x \rightarrow 5, y \rightarrow 7 \rangle$.

Vous pouvez faire cela sur l'arbre représentant l'expression.

Exercice 4 : Affectations

Étant donné un environnement $\rho = \langle x \rightarrow 4, y \rightarrow 5 \rangle$.

On considère les instructions suivantes:

- $I1 = \text{Affect}(x, \text{Binop}(\text{Add}, y, 2))$,
- $I2 = \text{Affect}(y, \text{Binop}(\text{Sub}, x, 3))$.

Que vaut $\llbracket I1 \rrbracket(\rho)$?

Que vaut $\llbracket I2 \rrbracket(\rho)$?

Que vaut $\llbracket \text{Block}([I1; I2]) \rrbracket(\rho)$?

Que vaut $\llbracket \text{Block}([I2; I1]) \rrbracket(\rho)$?

Exercice 5 : IfThenElse

On considère l'instruction suivante :

$I = \text{IfThenElse}(\text{Binop}(\text{Eq}, x, 2), \text{Affect}(y, 4), \text{Affect}(z, 5))$

On considère l'environnement $\rho_1 = \langle x \rightarrow 2, y \rightarrow 0, z \rightarrow 0 \rangle$.

Que vaut $\llbracket I \rrbracket(\rho_1)$?

On considère l'environnement $\rho_2 = \langle x \rightarrow 3, y \rightarrow 0, z \rightarrow 0 \rangle$.

Que vaut $\llbracket I \rrbracket(\rho_2)$?

Exercice 6 : While

On considère l'instruction suivante :

$I = \text{While}(\text{Binop}(\text{Neq}, x, 0), B)$, où B est une instruction dont on admettra que la sémantique est $\llbracket B \rrbracket(\rho) = \rho[x \leftarrow [x]_\rho + i, y \leftarrow [y]_\rho - i]$.

On considère l'environnement $\rho_1 = \langle x \rightarrow 0, y \rightarrow 1 \rangle$.

Que vaut $\llbracket I \rrbracket(\rho_1)$?

On considère l'environnement $\rho_2 = \langle x \rightarrow -2, y \rightarrow 0 \rangle$.

Que vaut $\llbracket I \rrbracket(\rho_2)$?

On considère l'environnement $\rho_3 = \langle x \rightarrow 2, y \rightarrow 2 \rangle$.

Que vaut $\llbracket I \rrbracket(\rho_3)$?

Exercice 7 : Tableaux

On considère l'instruction suivante :

$I1 = \text{AffectArray}(t, 0, \text{Binop}(\text{Add}, \text{ArrayVal}(t, 1), \text{ArraySize}(t)))$.

On considère l'environnement $\rho_1 = \langle t \rightarrow \text{array}(t\#, \rho_1), t\#0 \rightarrow 0, t\#1 \rightarrow 4, t\#size \rightarrow 4 \rangle$.

Que vaut $\llbracket I1 \rrbracket(\rho_1)$?

On considère $\rho_2 = \langle t \rightarrow \text{array}(t\#, \rho_2), t\#0 \rightarrow 0, t\#1 \rightarrow 4, t\#size \rightarrow 5 \rangle$.

Que vaut $\llbracket I1 \rrbracket(\rho_2)$?

On considère $\rho_3 = \langle t \rightarrow \text{array}(t\#, \rho_3), t\#0 \rightarrow 34, t\#1 \rightarrow 21, t\#size \rightarrow 0 \rangle$

Que vaut $\llbracket I1 \rrbracket(\rho_3)$? (décrivez l'effet global sur la mémoire en listant le résultat des trois environnements)

On considère l'instruction suivante :

$I2 = \text{DeclArray}(t, 12)$.

Que vaut $\llbracket I2 \rrbracket(\rho_2)$?

Que vaut $\llbracket \text{Block}([I2; I1]) \rrbracket(\rho_2)$?

Exercice 8 : Fonction

On considère l'instruction suivante :

$I = \text{Block}[\text{Affect}(x, \text{Binop}(\text{Mul}, x, x)); \text{Affect}(y, \text{Binop}(\text{Sub}, y, 1))]$.

On admettra que la sémantique est $\llbracket I \rrbracket(\rho) = \rho_1[y \leftarrow [y]_{\rho_1} - 1]$, avec $\rho_1 = \rho[x \leftarrow [x]_{\rho} * [x]_{\rho}]$.

On considère l'instruction $R = \text{Return_expr}(\text{Binop}(\text{Add}, x, y))$.

Étant donné un environnement $\rho = \langle x \rightarrow 4, y \rightarrow 7 \rangle$:

Que vaut $\llbracket R \rrbracket(\rho)$?

Que vaut $\llbracket \text{Block}([I; R]) \rrbracket(\rho)$?

Que vaut $\llbracket \text{Block}([R; I]) \rrbracket(\rho)$?

On considère maintenant les déclarations de fonctions suivantes :

- $f1 = \text{DeclF}(\text{int}, f1, [(\text{Value}, \text{int}, x); (\text{Value}, \text{int}, y)], \text{Block}([I; R]))$
- $f2 = \text{DeclF}(\text{int}, f2, [(\text{Reference}, \text{int}, x); (\text{Value}, \text{int}, y)], \text{Block}([I; R]))$
- $f3 = \text{DeclF}(\text{int}, f3, [(\text{Value}, \text{int}, x); (\text{Reference}, \text{int}, y)], \text{Block}([I; R]))$
- $f4 = \text{DeclF}(\text{int}, f4, [(\text{Reference}, \text{int}, x); (\text{Reference}, \text{int}, y)], \text{Block}([I; R]))$

On considère que ces déclarations ont été évaluées dans l'environnement des déclarations de fonctions (qui reste implicite), et on considère l'environnement suivant :

$\rho_2 = \langle a \rightarrow 4, b \rightarrow 3, t \rightarrow \text{array}(t\#, \rho_2), t\#size \rightarrow 1, t\#0 \rightarrow 2 \rangle$.

On considère les instructions suivantes, pour i entre 1 et 4 :

$J_i = \text{Proc}(f_i, [a; b])$.

Pour chacune d'entre elles, calculez $\llbracket J_i \rrbracket(\rho_2)$, et précisez la valeur de retour (qu'on n'utilise pas). On pourra se resservir du début de l'exercice.

On considère maintenant les expressions :

$K_i = \text{Func}(f_i, [\text{Binop}(\text{Add}, a, 4); \text{ArrayVal}(t, 0)])$.

Pour chacune, donnez $\llbracket K_i \rrbracket_{\rho_2}$ si elle est définie (en précisant les éventuels effets de bord sur ρ_2).

On considère maintenant l'instruction $L = \text{Proc}(f4, a, a)$. Calculez $\llbracket L \rrbracket(\rho_2)$, et précisez la valeur de retour.