Compilation et langages

TD sémantique et transformations de boucles

Hodor : un langage qui tourne en boucle (première partie)

On s'intéresse à un langage impératif minimal, Hodor, dans lequel on manipule des variables et des valeurs entières. On dispose d'une instruction conditionnelle `if(x) { ... }` qui exécute un bloc de code si la valeur de la variable `x` est non nulle, et d'une boucle `while(x) { ... }` dont le corps est exécuté tant que la valeur de la variable `x` est non nulle.

Un programme Hodor est un bloc *block* construit avec la grammaire suivante :

block	::=	*instr**	bloc d'instructions
instr	::=	*x* `=` *atom*	affectation simple
	|	*x* `=` *atom binop atom*	opération
	|	`if(` *x* `) {` *block* `} else {` *block* `}`	branchement conditionnel
	|	`while(` *x* `) {` *block* `}`	boucle conditionnelle
	|	`print(` *x* `)`	affichage
atom	::=	*n*	constante entière
	|	*x*	variable
binop	::=	`+` | `-` | `*` | `/`	

Par exemple, l'exécution du programme suivant affiche `64`.

```
x = 2
n = 6
if (x) {
  r = 1
  while (n) {
    r = x * r
    n = n - 1
} else {
  r = 0
}
print (r)
```

Sémantique

On donne pour les programmes Hodor une sémantique à grands pas basée sur les éléments suivants :

— Un état est une fonction S des variables vers les entiers.

— Les jugements $S \overset{p}{\underset{o}{\Longrightarrow}} S'$ et $S \overset{i}{\underset{o}{\longrightarrow}} S'$ signifient respectivement que le programme p et l'instruction i exécutés dans l'état S affichent la séquence d'entiers o et mènent à l'état S'.

Dans les règles ci-dessous, on utilise de plus les notations suivantes :

— $S[x \mapsto k]$ désigne l'état S' tel que $S'(x) = k$ et pour tout $y \neq x$, $S'(y) = S(x)$.

— $i :: p$ désigne un programme dont la première instruction est i est dont les instructions suivantes sont celles de p.

— $[\![a]\!]_S$ désigne la valeur de l'atome a dans l'état S : $[\![n]\!]_S = n$ et $[\![x]\!]_S = S(x)$; cette notation est étendue aux opérations binaires, avec par exemple $[\![a_1 \texttt{ + } a_2]\!]_S = [\![a_1]\!]_S + [\![a_2]\!]_S$.

$$\frac{}{S \xRightarrow[\emptyset]{\emptyset} S} \qquad\qquad \frac{S_0 \xrightarrow[o_1]{i} S_1 \qquad S_1 \xRightarrow[o_2]{p} S_2}{S_0 \xRightarrow[o_1,o_2]{i\ ::\ p} S_2}$$

$$\frac{[\![e]\!]_S = k}{S \xrightarrow[k]{\texttt{print}(x)} S} \qquad\qquad \frac{[\![e]\!]_S = k}{S \xrightarrow[\emptyset]{x\ \texttt{=}\ e} S[x \mapsto k]}$$

$$\frac{S_0(x) \neq 0 \qquad S_0 \xRightarrow[o]{p_1} S_1}{S_0 \xrightarrow[o]{\texttt{if}(x)\ \{p_1\}\ \texttt{else}\ \{p_2\}} S_1} \qquad\qquad \frac{S_0(x) = 0 \qquad S_0 \xRightarrow[o]{p_2} S_1}{S_0 \xrightarrow[o]{\texttt{if}(x)\ \{p_1\}\ \texttt{else}\ \{p_2\}} S_1}$$

$$\frac{S(x) = 0}{S \xrightarrow[\emptyset]{\texttt{while}(x)\ \{p\}} S} \qquad\qquad \frac{S_0(x) \neq 0 \qquad S_0 \xRightarrow[o_1]{p} S_1 \qquad S_1 \xrightarrow[o_2]{\texttt{while}(x)\ \{p\}} S_2}{S_0 \xrightarrow[o_1,o_2]{\texttt{while}(x)\ \{p\}} S_2}$$

Question 1. Pour les deux programmes ci-dessous, déterminer o et S tels que $\emptyset \xRightarrow[o]{p_i} S$.

```
x = 0
if (x) {
  print (x)
} else {
  x = x + 7
  print (x)
}
```

```
x = 2
r = 1
while (x) {
  r = r + r
  x = x - 1
}
print (r)
```

On dit que deux programmes Hodor p_1 et p_2 sont équivalents lorsque, pour tous états S et S' et pour toute séquence d'entiers o, $S \xRightarrow[o]{p_1} S'$ si et seulement si $S \xRightarrow[o]{p_2} S'$.

On veut démontrer que pour toute variable x et pour tout bloc d'instructions p, les deux programmes ci-dessous sont équivalents :

```
while (x) {
  p
}
```

```
if (x) {
  p
  while (x) {
    p
  }
} else {}
```

Pour tous deux programmes p_1 et p_2, on note $p_1 @ p_2$ le programme formé par la concaténation des instructions de p_1 et p_2.

Question 2. Démontrer que si $S_0 \xRightarrow[o_1]{p_1} S_1$ et $S_1 \xRightarrow[o_2]{p_2} S_2$ alors $S_0 \xRightarrow[o_1,o_2]{p_1 @ p_2} S_2$.

Indice : raisonner par récurrence sur la dérivation de $S_0 \xRightarrow[o_1]{p_1} S_1$.

Question 3. Démontrer que si $S \xrightarrow[o]{\texttt{while}(x)\ \{p\}} S'$ alors $S \xrightarrow[o]{\texttt{if}(x)\ \{\ p\ @\ \texttt{while}(x)\{p\}\ \}\ \texttt{else}\ \{\}} S'$.

Question 4 (*Bonus*)**.** Démontrer que si $S \xrightarrow[o]{\texttt{if}(x)\ \{p\ @\ \texttt{while}(x)\{p\}\}\ \texttt{else}\ \{\}} S'$ alors $S \xrightarrow[o]{\texttt{while}(x)\ \{p\}} S'$.

Indice : vous avez besoin d'un lemme sur la concaténation @.