Sémantique des langages de programmation
Examen final

Semestre 5, 2013–14

Durée : 1h45.
Tout document personnel autorisé (pas de prêt entre voisins).
Ce sujet comporte 4 exercices indépendants, qui peuvent être traités dans l'ordre voulu.
Il contient 3 pages. Il va de soi que toute réponse devra être justifiée.

Exercice 1 : Équivalence de programmes

Montrer l'équivalence ou la non-équivalence des programmes IMP suivants :

1. `x := 10; while not (x = 0) do x := x - 1` et `x := 0`
2. `while not (x = 0) do x := x - 1` et `x := 0`
3. `if a then b else c` et `if not a then c else b`
4. `if a and b then c else d` et `if a then (if b then c else d) else d`
5. `if not (a and b) then c else d` et `if (not b) or (not a) then c else d`

Exercice 2 : Expressions booléennes

On rappelle dans la figure 1 la sémantique opérationnelle à grand pas des expressions booléennes vue en cours.

1. Proposer une sémantique opérationnelle à petit pas pour les expressions booléennes.
2. Grâce à celle-ci, donner la sémantique de l'expression `(x <= 3 and x > 0) or x + x <= 10` dans l'environnement $\sigma = \{\mathtt{x} \mapsto 1\}$.
3. Montrer l'équivalence de votre sémantique opérationnelle à petit pas avec la sémantique opérationnelle à grand pas du cours.

$$(B_1)\frac{}{\langle t, \sigma\rangle \rightsquigarrow t} \quad t \in \{\mathsf{true}; \mathsf{false}\} \qquad (B_{14})\frac{\langle b, \sigma\rangle \rightsquigarrow t}{\langle \mathsf{not}\ b, \sigma\rangle \rightsquigarrow (\neg t)}$$

$$(B_2)\frac{\langle a_1, \sigma\rangle \rightsquigarrow n_1 \quad \langle a_2, \sigma\rangle \rightsquigarrow n_2}{\langle a_1 = a_2, \sigma\rangle \rightsquigarrow (n_1 = n_2)} \quad (B_3)\frac{\langle a_1, \sigma\rangle \rightsquigarrow n_1 \quad \langle a_2, \sigma\rangle \rightsquigarrow n_2}{\langle a_1 \le a_2, \sigma\rangle \rightsquigarrow (n_1 \le n_2)} \quad \begin{array}{l} n_1, n_2 \in \mathbb{Z} \\ a_1, a_2 \in E_A \end{array}$$

$$(B_8)\frac{\langle b_1, \sigma\rangle \rightsquigarrow \mathsf{true} \quad \langle b_2, \sigma\rangle \rightsquigarrow v}{\langle b_1\ \mathsf{and}\ b_2, \sigma\rangle \rightsquigarrow v} \quad (B_9)\frac{\langle b_1, \sigma\rangle \rightsquigarrow \mathsf{false}}{\langle b_1\ \mathsf{and}\ b_2, \sigma\rangle \rightsquigarrow \mathsf{false}}$$

$$(B_{11})\frac{\langle b_1, \sigma\rangle \rightsquigarrow \mathsf{false} \quad \langle b_2, \sigma\rangle \rightsquigarrow v}{\langle b_1\ \mathsf{or}\ b_2, \sigma\rangle \rightsquigarrow v} \quad (B_{12})\frac{\langle b_1, \sigma\rangle \rightsquigarrow \mathsf{true}}{\langle b_1\ \mathsf{or}\ b_2, \sigma\rangle \rightsquigarrow \mathsf{true}}$$

FIGURE 1: Sémantique opérationnelle à grand pas des expressions booléennes

$$(FS_{v1})\frac{}{(\texttt{fun}\ x\ \texttt{->}\ e)v \hookrightarrow_v \{v/x\}e}$$

$$(FS_{v2})\frac{e_1 \hookrightarrow_v e_1'}{e_1\ e_2 \hookrightarrow_v e_1'\ e_2} \qquad (FS_{v3})\frac{e_2 \hookrightarrow_v e_2'}{v\ e_2 \hookrightarrow_v v\ e_2'}$$

$$(FS_{v4})\frac{}{v_1 + v_2 \hookrightarrow_v v_1 [\![+]\!] v_2}$$

$$(FS_{v5})\frac{e_1 \hookrightarrow_v e_1'}{e_1 + e_2 \hookrightarrow_v e_1' + e_2} \qquad (FS_{v6})\frac{e_2 \hookrightarrow_v e_2'}{v + e_2 \hookrightarrow_v v + e_2'}$$

FIGURE 2: Sémantique opérationnelle à petit pas par valeur de MiniML

Exercice 3 : Logique de Hoare

1. Montrer que le triplet de Hoare $\{x \leq 2\}$if x $<$ 0 then x := 2 else skip$\{x \geq 0\}$ est valide.
2. On considère le programme `p` suivant :
   `y := x; while not (x = 0) do (x := x - 1; y := y + 1)`
   a) Montrer que $n - x = y - n$ est un invariant de la boucle.
   b) Montrer que $\{x = n\}$p$\{y = 2n\}$ est valide.
3. On considère le programme `sum` annoté :

```
i := 0;
s := 0;
/*@ loop invariant s = i * i; */
while not (i = n) do (
  i := i + 1;
  s := s + 2 * i - 1 )
```

   Calculer $VC(\texttt{sum}, s = n * n)$. En déduire que $\{\texttt{true}\}\texttt{sum}\{s = n * n\}$ est valide.

Exercice 4 : Réductions parallèles

On étend la syntaxe du langage MiniML vu en cours avec la possibilité de mettre des expressions en parallèle :

$$e ::= n \mid x \mid \texttt{fun}\ x\ \texttt{->}\ e \mid e\ e \mid e + e \mid \boldsymbol{e \wr e}$$

Dans un premier temps, on étend la sémantique opérationnelle à petit pas par valeur (cf. figure 2) uniquement avec les règles suivantes :

$$(FS_{v7})\frac{e_1 \hookrightarrow_v e_1'}{e_1 \wr e_2 \hookrightarrow_v e_1' \wr e_2} \qquad (FS_{v8})\frac{e_2 \hookrightarrow_v e_2'}{e_1 \wr e_2 \hookrightarrow_v e_1 \wr e_2'}$$

$$(F_1)\frac{}{n \rightarrow_v n} \qquad (F_2)\frac{}{\texttt{fun } x \texttt{ -> } e \rightarrow_v \texttt{fun } x \texttt{ -> } e}$$

$$(F_{v3})\frac{e_1 \rightarrow_v \texttt{fun } x \texttt{ -> } e \qquad e_2 \rightarrow_v v \qquad \{v/x\}e \rightarrow_v v'}{e_1\ e_2 \rightarrow_v v'}$$

$$(F_4)\frac{e_1 \rightarrow_v v_1 \qquad e_2 \rightarrow_v v_2}{e_1 + e_2 \rightarrow_v v_1 [\![+]\!] v_2}$$

Figure 3: Sémantique opérationnelle à grand pas par valeur de MiniML

On redéfinit une valeur comme étant une variable, une abstraction, ou la mise en parallèle de deux valeurs. Les valeurs sont donc décrites par la grammaire :

$$v ::= x \mid \texttt{fun } x \texttt{ -> } e \mid v \wr v$$

1. Donner la sémantique des expressions suivantes en détaillant les séquences de calcul :
   - $((\texttt{fun } x \texttt{ -> } x + 3)\ 4) \wr (6 + 2)$
   - $(\texttt{fun } f \texttt{ -> } (f \wr 3))\ ((3 + 2) \wr 1)$
2. Proposer une extension de la sémantique opérationnelle à grand pas par valeur (cf. figure 3), et montrer qu'elle est équivalente à la sémantique opérationnelle à petit pas. Indication : il pourra être utile de montrer le lemme suivant : si $e_1 \wr e_2 \hookrightarrow_v^n v$ alors $v = v_1 \wr v_2$ et $e_1 \hookrightarrow_v^{n_1} v_1$ et $e_2 \hookrightarrow_v^{n_2} v_2$ avec $n = n_1 + n_2$.
3. Montrer que la sémantique reste déterministe : si $e \hookrightarrow_v^* v_1$ et $e \hookrightarrow_v^* v_2$ où v_1 et v_2 sont des valeurs, alors $v_1 = v_2$. Indication : il peut être judicieux de se placer dans le cadre de la sémantique à grand pas et d'utiliser le résultat d'équivalence de la question précédente.
4. L'expression $(\texttt{fun } f \texttt{ -> } f\ 1)\ ((\texttt{fun } x \texttt{ -> } x) \wr (\texttt{fun } x \texttt{ -> } -x))$ possède-t-elle une sémantique ?

On ajoute la règle suivante :

$$(Dist)\frac{}{f\ (e_1 \wr e_2) \hookrightarrow_v (f\ e_1) \wr (f\ e_2)}$$

5. Donner la sémantique des expressions suivantes en détaillant les réductions :
   - $(\texttt{fun } x \texttt{ -> } x)\ (3 + 4 \wr 5)$
   - $(\texttt{fun } f \texttt{ -> } f\ 2)\ ((\texttt{fun } x \texttt{ -> } x + 1) \wr (\texttt{fun } x \texttt{ -> } x + x))$
6. Montrer que la sémantique n'est plus déterministe. On pourra considérer le terme $(\texttt{fun } x \texttt{ -> } (\texttt{fun } x \texttt{ -> fun } y \texttt{ -> } x + y)\ x\ x)\ (1 \wr 2)$.