

TD 6

Exercice 1. Prouvez que pour tout environnement ρ , il n'existe pas d'environnement ρ' tel que $\rho \vdash \text{while } \dot{\text{I}} \text{ do skip} \Rightarrow \rho'$.

Exercice 2. Étant donnés deux programmes c_1, c_2 , on dit que c_1 et c_2 sont équivalents, noté $c_1 \sim c_2$ ssi pour tout environnements ρ, ρ' , $(\rho \vdash c_1 \Rightarrow \rho') \Leftrightarrow (\rho \vdash c_2 \Rightarrow \rho')$

1. Montrer que $\text{while } e \text{ do } c \sim \text{if } e \text{ then } (c; \text{while } e \text{ do } c) \text{ else skip}$
2. Quels sont les programmes équivalents à $\text{while } \dot{\text{I}} \text{ do skip}$?

Exercice 3. L'un des objectifs des cours suivants sera d'introduire les outils mathématiques nécessaires pour pouvoir définir une sémantique dénotationnelle de IMP : $\llbracket c \rrbracket \rho = \rho'$, ce qui signifie que l'exécution du programme c dans l'environnement ρ mène à l'environnement ρ' . Cette écriture « fonctionnelle » suggère que l'environnement ρ' est entièrement déterminé par c et ρ (bien sûr, puisque nos programmes sont « déterministes »).

1. Montrer cette propriété pour la sémantique opérationnelle : pour tout c, ρ, ρ_1, ρ_2 , si $\rho \vdash c \Rightarrow \rho_1$ et $\rho \vdash c \Rightarrow \rho_2$ alors $\rho_1 = \rho_2$.
2. On considère le langage non déterministe donné par la syntaxe suivante :

$$c ::= \text{skip} \mid x := e \mid c; c \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c \mid c \vee c$$

où l'instruction $c_1 \vee c_2$ signifie « exécute c_1 ou exécute c_2 , de manière non déterministe ».

Proposez une sémantique opérationnelle de ce langage.

Exercice 4. En cours, vous avez aperçu la distinction entre sémantique dénotationnelle et sémantique opérationnelle des programmes IMP. Cependant, vous n'avez utilisé qu'une sémantique dénotationnelle des expressions arithmétiques (rappelée ci dessous). Dans cet exercice, on s'étudie le cas de deux extensions des opérations arithmétiques :

1. On étend nos expressions arithmétiques par la syntaxe suivante :

$$e ::= x \mid \dot{n} \mid e \dot{+} e \mid \dot{-} e \mid \dot{f}(e)$$

Pour interpréter le symbole \dot{f} , on suppose disposer d'une fonction **partielle** f des entiers dans les entiers.

- (a) Donnez une sémantique opérationnelle à grands pas (inspirez vous de celle pour IMP, vue en cours et rappelée en figure 1) pour ces expressions.
 - (b) Étendez la sémantique dénotationnelle vue en cours pour ces expressions.
 - (c) Montrez l'équivalence des deux sémantiques.
 - (d) Quelle différence essentielle rend la sémantique dénotationnelle pour les expressions arithmétiques beaucoup plus simple que celle pour les programmes?
2. Expressions arithmétiques avec effets de bords : on étend nos expressions arithmétiques par la syntaxe suivante :

$$e ::= x \mid \dot{n} \mid e \dot{+} e \mid \dot{-} e \mid c \text{ resultis } e$$

Intuitivement, pour évaluer l'expression $c \text{ resultis } e$, on évalue d'abord la commande c , puis on évalue e dans l'environnement obtenu.

$$\begin{array}{c}
\frac{}{\rho \vdash x := e \Rightarrow \rho[x \mapsto \llbracket e \rrbracket \rho]} \text{ (:=)} \qquad \frac{}{\rho \vdash \text{skip} \Rightarrow \rho} \text{ (Skip)} \\
\\
\frac{\rho \vdash c_1 \Rightarrow \rho' \quad \rho' \vdash c_2 \Rightarrow \rho''}{\rho \vdash c_1; c_2 \Rightarrow \rho''} \text{ (Seq)} \\
\\
\frac{\rho \vdash c_1 \Rightarrow \rho'}{\rho \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 \Rightarrow \rho' \text{ si } \llbracket e \rrbracket \rho \neq 0} \text{ (if}_1\text{)} \qquad \frac{\rho \vdash c_2 \Rightarrow \rho'}{\rho \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 \Rightarrow \rho' \text{ si } \llbracket e \rrbracket \rho = 0} \text{ (if}_2\text{)} \\
\\
\frac{\rho \vdash c \Rightarrow \rho' \quad \rho' \vdash \text{while } e \text{ do } c \Rightarrow \rho''}{\rho \vdash \text{while } e \text{ do } c \Rightarrow \rho'' \text{ si } \llbracket e \rrbracket \rho \neq 0} \text{ (while)} \qquad \frac{}{\rho \vdash \text{while } e \text{ do } c \Rightarrow \rho \text{ si } \llbracket e \rrbracket \rho = 0} \text{ (while}_{\text{fin}}\text{)}
\end{array}$$

FIGURE 1 – La sémantique opérationnelle à grands pas de IMP.

- Donnez des sémantiques opérationnelle et dénotationnelle pour ces expressions (on supposera disposer d'une sémantique pour les programmes c).
- Comment s'évalue le terme $((x := x + 1) \text{ resultis } x) + ((y := x + x) \text{ resultis } y)$ dans l'environnement $\rho = [x \mapsto 3]$.
- Vous comprenez désormais les horreurs que permet d'écrire le $C : T[i++] = i++$.

Exercice 5. Dans cet exercice, on s'intéresse à la sémantique opérationnelle à petits pas de IMP, rappelée en figure 2.

- Progrès : Montrez que les états terminaux, c'est à dire les couples (C, ρ) tels que pour tous couples (C', ρ') , $(C, \rho) \not\rightarrow (C', \rho')$, sont **exactement** les couples de la forme (ε, ρ) .
- Déterminisme : Montrez que pour tout état (C, ρ) , il existe au plus un état (C', ρ') tel que $(C, \rho) \rightarrow (C', \rho')$.
- Non Terminaison : Donnez un exemple de programme c tel que pour tout environnement ρ et pour tout état final (ε, ρ') , $(c \cdot \varepsilon, \rho) \not\rightarrow^* (\varepsilon, \rho')$.

$$\begin{array}{ll}
(x := e \cdot C, \rho) \rightarrow (C, \rho[x \mapsto \llbracket e \rrbracket \rho]) & (1) \\
(\text{skip} \cdot C, \rho) \rightarrow (C, \rho) & (2) \\
(c_1; c_2 \cdot C, \rho) \rightarrow (c_1 \cdot c_2 \cdot C, \rho) & (3) \\
(\text{if } e \text{ then } c_1 \text{ else } c_2 \cdot C, \rho) \rightarrow (c_1 \cdot C, \rho) \quad \text{si } \llbracket e \rrbracket \rho \neq 0 & (4) \\
(\text{if } e \text{ then } c_1 \text{ else } c_2 \cdot C, \rho) \rightarrow (c_2 \cdot C, \rho) \quad \text{si } \llbracket e \rrbracket \rho = 0 & (5) \\
(\text{while } e \text{ do } c \cdot C, \rho) \rightarrow (c \cdot \text{while } e \text{ do } c \cdot C, \rho) \quad \text{si } \llbracket e \rrbracket \rho \neq 0 & (6) \\
(\text{while } e \text{ do } c \cdot C, \rho) \rightarrow (C, \rho) \quad \text{si } \llbracket e \rrbracket \rho = 0 & (7)
\end{array}$$

FIGURE 2 – La sémantique opérationnelle à petits pas de IMP.

Exercice 6. On s'intéresse à une sémantique opérationnelle des expressions arithmétiques dite à petits pas (cf. figure 3).

$$\begin{array}{c}
\frac{}{(x, \rho) \rightarrow_{pp} (\widehat{\rho(x)}, \rho)} \text{ (Var)} \qquad \frac{}{(\dot{n}, \rho) \rightarrow_{pp} n} \text{ (Nb)} \\
\frac{(e_1, \rho) \rightarrow_{pp} (e'_1, \rho)}{(e_1 \dot{+} e_2, \rho) \rightarrow_{pp} (e'_1 \dot{+} e_2, \rho)} \text{ (+}_\ell\text{)} \qquad \frac{(e_2, \rho) \rightarrow_{pp} (e'_2, \rho)}{(\dot{n} \dot{+} e_2, \rho) \rightarrow_{pp} (\dot{n} \dot{+} e'_2, \rho)} \text{ (+}_r\text{)} \\
\frac{}{(\dot{n} \dot{+} \dot{m}, \rho) \rightarrow_{pp} (\widehat{\dot{n} \dot{+} \dot{m}}, \rho)} \text{ (+}_{\text{fin}}\text{)} \qquad \frac{(e, \rho) \rightarrow_{pp} (e'', \rho'') \quad (e'', \rho'') \rightarrow_{pp} (e', \rho')}{(e, \rho) \rightarrow_{pp} (e', \rho')} \text{ (Tr)} \\
\frac{(e, \rho) \rightarrow_{pp} (e', \rho)}{(\dot{-}e, \rho) \rightarrow_{pp} (\dot{-}e', \rho)} \text{ (-)} \qquad \frac{}{(\dot{-}\dot{n}, \rho) \rightarrow_{pp} (\widehat{\dot{-}\dot{n}}, \rho)} \text{ (-}_{\text{fin}}\text{)}
\end{array}$$

FIGURE 3 – Sémantique opérationnelle à petits pas des expressions arithmétiques.

1. Donnez une preuve que $((x \dot{+} (\dot{-}y)) \dot{+} \dot{2}, \rho[x \mapsto 3, y \mapsto 1]) \rightarrow_{pp} 4$.
2. Montrez que les règles ont les propriétés suivantes :
 - (a) Progrès : les états terminaux (ceux pour lesquels aucune règle ne s'appliquent) sont de la forme (\dot{n}, ρ) où n est un entier.
 - (b) Déterminisme : étant donné une expression e et un environnement ρ , il existe au plus un couple (e', ρ') tel que $(e, \rho) \rightarrow_{pp} (e', \rho')$