TD 9

Exercice 1. On considère le langage $\{*, \bullet\}^*$ équipé de la sémantique à petits pas suivante :

$$\begin{array}{ll} X * \bullet Y \to XY & \text{si } \exists n \geq 0 \,.\, X = *^n \\ X \bullet * Y \to XY & \text{si } \exists n \geq 0 \,.\, X = \bullet^n \end{array}$$

1. Montrez que cette sémantique est déterministe, *i.e.* si $A \to B$ et $A \to B'$ alors $B = B'$.
2. On considère le DCPO (non pointé) $\{0, 1\}$ équipé de l'ordre discret (« plat »), et on se donne la sémantique définie par :

$$\begin{array}{rl} [\![\varepsilon]\!]_2 = 0; & \\ [\![aX]\!]_2 = 1 - [\![X]\!]_2 & a \in \{*, \bullet\} \end{array}$$

   Montrez que cette sémantique est correcte par rapport à la sémantique à petits pas.
3. Même question pour le DCPO (non pointé) des entiers relatifs équipés de l'ordre plat et la sémantique suivante :

$$\begin{aligned} [\![\varepsilon]\!]_{\mathbb{Z}} &= 0 \\ [\![*X]\!]_{\mathbb{Z}} &= 1 + [\![X]\!]_{\mathbb{Z}} \\ [\![\bullet X]\!]_{\mathbb{Z}} &= -1 + [\![X]\!]_{\mathbb{Z}} \end{aligned}$$

4. On se donne la notion d'équivalence observationnelle suivante :

   $A \simeq B$ lorsque pour tout contexte $C[.]$, $C[A] \to^* \varepsilon$ ssi $C[B] \to^* \varepsilon$.

   Montrez qu'il s'agit d'une relation d'équivalence. Les deux sémantiques dénotationnelles ci-dessus sont-elles complètement abstraites pour cette équivalence observationnelle ?

Exercice 2. On considère le langage Imp défini par la grammaire

$$\begin{array}{ll} e ::= x \mid 0 \mid 1 \mid e + e \mid -e \mid e \times e & \text{(expressions)} \\ b ::= e = e \mid \neg b \mid b \wedge b & \text{(conditions)} \\ c ::= \textbf{skip} \mid x := e \mid c; c \mid \textbf{if } b \textbf{ then } c \textbf{ else } c \textbf{ fi} \mid \textbf{while } b \textbf{ do } c & \text{(instructions)} \end{array}$$

où x est une variable de programme, et la logique $\mathrm{FO}[0, 1, +, \times]$ (une variante de l'arithmétique de Robinson) définie par la grammaire

$$\begin{array}{ll} t ::= e \mid i & \text{(termes logiques)} \\ \varphi ::= t = t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists i.\varphi & \text{(formules logiques)} \end{array}$$

où i est une variable logique à valeur entière.

1. La grammaire ci-dessus étend légèrement le langage Imp vu en cours, en permettant d'utiliser des *conditions* d'instructions conditionnelles **if** ou d'instructions d'itération **while**. On étend de ce fait la sémantique dénotationnelle des expressions pour traiter les conditions :

$$[\![e_1 = e_2]\!]\rho = \begin{cases} 0 & \text{si } [\![e_1]\!]\rho \neq [\![e_2]\!]\rho \\ 1 & \text{sinon} \end{cases} \quad ; \quad [\![\neg b]\!]\rho = 1 - [\![b]\!]\rho \,; \quad [\![b_1 \wedge b_2]\!]\rho = [\![b_1]\!]\rho \times [\![b_2]\!]\rho \,.$$

Par ailleurs, il nous faut une sémantique pour les *formules logiques*. On définit pour une interprétation $\rho \in \mathit{Env}$ la relation de satisfaction $\rho \models \varphi$ comme attendue :

$$\begin{aligned}
&\rho \models t_1 = t_2 && \text{si } [\![t_1]\!]\rho = [\![t_2]\!]\rho\,;\\
&\rho \models \neg\varphi && \text{si } \rho \not\models \varphi\,;\\
&\rho \models \phi \wedge \psi && \text{si } \rho \models \varphi \text{ et } \rho \models \psi\,;\\
&\rho \models \exists i.\varphi && \text{si } \rho \models \varphi[i := n] \text{ pour un certain } n\,.
\end{aligned}$$

Syntaxiquement, une expression e de Imp est aussi un terme logique, et par suite une condition b de Imp est aussi une formule logique. Montrer que les deux sémantiques sont équivalentes : pour toute condition b et tout environnement ρ, $[\![b]\!]\rho \neq 0$ si et seulement si $\rho \models b$.

2. On reprend la sémantique dénotationelle de Imp $[\![c]\!] : \mathit{Env} \to \mathit{Env}_\perp$ vue en en cours :

$$\begin{aligned}
[\![x := e]\!]\rho &= \rho[x \mapsto [\![e]\!]\rho]\,;\\
[\![\mathbf{skip}]\!]\rho &= \rho\,;\\
[\![c_1; c_2]\!]\rho &= \begin{cases} \perp & \text{si } [\![c_1]\!]\rho = \perp\,,\\ [\![c_2]\!]([\![c_1]\!]\rho) & \text{sinon}\,; \end{cases}\\
[\![\mathbf{if}\ b\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2\ \mathbf{fi}]\!]\rho &= \begin{cases} [\![c_1]\!]\rho & \text{si } [\![b]\!]\rho \neq 0\,,\\ [\![c_2]\!]\rho & \text{sinon}\,; \end{cases}\\
[\![\mathbf{while}\ b\ \mathbf{do}\ c]\!]\rho &= \mathrm{lfp}(F_{b,c})(\rho)\,,
\end{aligned}$$

où $F_{b,c} : [\mathit{Env} \to \mathit{Env}_\perp] \to [\mathit{Env} \to \mathit{Env}_\perp]$ est définie par :

$$F_{b,c}(f)(\rho) = \begin{cases} \rho & \text{si } [\![b]\!]\rho = 0\,,\\ \perp & \text{si } [\![b]\!]\rho \neq 0 \text{ et } [\![c]\!]\rho = \perp\,,\\ f([\![c]\!]\rho) & \text{sinon.} \end{cases}$$

On appelle *triplet de* Hoare un triplet $\{\varphi\}\ c\ \{\psi\}$. On dit qu'un triplet de Hoare $\{\varphi\}\ c\ \{\psi\}$ est *valide*, noté $\models \{\varphi\}\ c\ \{\psi\}$, si pour tout ρ,

$$(\rho \models \varphi \wedge [\![c]\!]\rho \neq \perp) \Rightarrow [\![c]\!]\rho \models \psi\,.$$

On introduit les règles de déduction suivantes :

$$\frac{}{\{\varphi\}\ \mathbf{skip}\ \{\varphi\}} \qquad \frac{}{\{\varphi[x := e]\}\ x := e\ \{\varphi\}} \qquad \frac{\{\varphi \wedge b\}\ c_1\ \{\psi\} \qquad \{\varphi \wedge \neg b\}\ c_2\ \{\psi\}}{\{\varphi\}\ \mathbf{if}\ b\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2\ \mathbf{fi}\ \{\psi\}}$$

$$\frac{\models \varphi \Rightarrow \varphi' \qquad \{\varphi'\}\ c\ \{\psi'\} \qquad \models \psi' \Rightarrow \psi}{\{\varphi\}\ c\ \{\psi\}}$$

Montrez que ce système de déduction est *correct* : tout triplet de Hoare prouvable est valide.

3. Proposez une règle pour le **while** et la composition séquentielle, et étendez le résultat de la question précédente au nouveau système.

4. À l'aide de ce système de preuve, donnez une preuve du triplet de Hoare

$$\{x = 0 \wedge y = 0 \wedge z = 0 \wedge n \geq 0\}\ c\ \{x = n^3\}$$

où c est le programme **while** $z < 3n$ **do** $z := z + 3$; $y := y + 2z - 3$; $x := x + y - z + 1$.

5. On appelle *plus faible précondition libérale* l'ensemble

$$\mathsf{wlp}(c, \varphi) := \{\rho \in \mathit{Env} \mid [\![c]\!](\rho) = \bot \text{ ou } [\![c]\!](\rho) \models \varphi\}\ .$$

Montrez que pour tout programme c sans boucle **while** et toute formule φ, il existe une formule $\mathsf{WLP}(c, \varphi)$ qui caractérise $\mathsf{wlp}(c, \varphi)$, c'est à dire que pour tout ρ, $\rho \models \mathsf{WLP}(c, \varphi)$ ssi $\rho \in \mathsf{wlp}(c, \varphi)$.

6. On admet que l'on peut définir une formule $\mathsf{WLP}(c, \varphi)$ pour tout programme c. En déduire que la logique de Hoare est *complète* : si le triplet $\{\varphi\}\ c\ \{\psi\}$ est valide, alors il est prouvable.
Indication : on pourra commencer par montrer que le triplet $\{\mathsf{WLP}(c, \psi)\}\ c\ \{\psi\}$ est prouvable.

7. On suppose fixé un système de preuve $\mathcal{S}$ de triplets de Hoare tel que
— $\mathcal{S}$ est correct : tous les triplets de Hoare prouvables sont valides.
— $\mathcal{S}$ est vérifiable : on peut décider, étant donné un arbre de preuve, si cet arbre de preuve est une preuve dans $\mathcal{S}$.
Montrez que $\mathcal{S}$ est incomplet.

8. Pourquoi la logique de Hoare est-elle malgré tout complète ?

9. On admet que la formule dont l'existence est admise en question 6 est de plus calculable. En déduire que le problème :
— Entrée : une formule φ
— Question : est-ce que φ est valide ?
n'est pas récursivement énumérable.

10. Prouvez la correction de l'exponentiation rapide.

Exercice 3. On étend le langage Imp avec deux nouvelles commandes **break** et **continue**. Un programme est dit bien formé si toute occurence de **break** et **continue** est à l'intérieur d'un while ; dans la suite, on suppose toujours les programmes bien formés.

On généralise la notion de triplet de Hoare à des triplets de la forme $\Pi \vdash \{\varphi\}\ c\ \{\psi\}$ où Π est une *pile de paires* (φ, ψ), et on remplace la règle du while par la règle suivante :

$$\frac{(\varphi, \psi) \cdot \Pi \vdash \{\varphi \wedge b\}\ c; \textbf{continue}\ \{\bot\} \qquad \varphi \wedge \neg b \models \psi}{\Pi \vdash \{\varphi\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{\psi\}}$$

1. Proposez deux règles pour **break** et **continue** de sorte que l'on puisse prouver le triplet de Hoare suivant :

$$\vdash \{y \geq 0\}\ \textbf{while}\ y > 0\ \textbf{do if}\ y = 1\ \textbf{then break else}\ y := y - 2\ \textbf{fi}\ \{y = 0 \vee y = 1\}$$

2. On appelle *continuation* une fonction $k : \mathit{Env} \to \mathit{Env}_\bot$ et on note K l'ensemble des continuations. Proposez une nouvelle sémantique dénotationelle $[\![c]\!]_{cps} : K \times (K \times K)^* \to K$ qui à une continuation k et à une pile de paires de continuations π associe la continuation $[\![c]\!]_{cps}(k, \pi)$.

3. On veut maintenant montrer que la logique de HOARE définie en première question est correcte. On introduit les définitions suivantes.
   — Une continuation k est *φ-sûre*, noté $\mathsf{safe}(k, \varphi)$, si pour tout environnement ρ, $\rho \models \varphi$ implique $k(\rho) = \bot$.
   — Une pile de paires de continuation $\pi = (k_{c,1}, k_{b,1}) \dots (k_{c,n}, k_{b,n})$ est *sûre* pour une pile de paire de formules $\Pi = (\varphi_1, \psi_1) \dots (\varphi_n, \psi_n)$, $\mathsf{safe}(\pi, \Pi)$, si pour tout $i = 1 \dots n$, $\mathsf{safe}(k_{c,i}, \varphi_i)$ et $\mathsf{safe}(k_{b,i}, \psi_i)$.
   — un triplet de HOARE $\Pi \vdash \{\varphi\}\ c\ \{\psi\}$ est *valide* si pour tout k, π, $\mathsf{safe}(k, \psi)$ et $\mathsf{safe}(\pi, \Pi)$ impliquent $\mathsf{safe}([\![c]\!]_{cps}(k, \pi), \varphi)$.

   Montrez que la logique de HOARE définie à la question 1 est correcte.