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. Le langage *PCF finitaire* est une variante de PCF où le type de base est celui des booléens et où il n'y a pas de points fixes. Plus précisément, les types sont définis par

$$\tau ::= \mathbf{bool} \mid \tau_1 \to \tau_2$$

et les termes par

$$M ::= x \mid M_1 M_2 \mid \lambda x : \tau . M \mid \mathbf{tt} \mid \mathbf{ff} \mid \bot \mid \mathbf{if}\ M_1\ \mathbf{then}\ M_2\ \mathbf{else}\ M_3\ \mathbf{fi}$$

où $\mathbf{tt}$, $\mathbf{ff}$, et $\bot$ sont de type $\mathbf{bool}$, et où la conditionnelle n'est définie que pour les N, P de type $\mathbf{bool}$. Les contextes d'évaluation sont définis par

$$C[.] ::= .M \mid \mathbf{if}\ .\ \mathbf{then}\ M\ \mathbf{else}\ N\ \mathbf{fi}$$

La sémantique *small-step* est définie par

$$\begin{array}{r} \bot \to \bot \\ (\lambda x : \tau . M) N \to M[N/x] \\ \mathbf{if\ tt\ then}\ M\ \mathbf{else}\ N\ \mathbf{fi} \to M \\ \mathbf{if\ ff\ then}\ M\ \mathbf{else}\ N\ \mathbf{fi} \to N \end{array}$$

avec la règle supplémentaire $C[M] \to C[M']$ si $M \to M'$ et si $C[.]$ est un contexte d'évaluation.

1. On dit qu'un terme M *converge* (noté $M \downarrow$) si il existe un terme N tel que $M \to^* N \not\to$. Donnez deux exemples de termes M de type $\mathbf{bool}$ tels que $M \not\downarrow$.
2. On définit la notion d'équivalence observationnelle suivante : $M \simeq N$ si pour tout contexte $C[.]$, $C[M] \to^* \mathbf{tt}$ ssi $C[N] \to^* \mathbf{tt}$. Montrer que $M \simeq N$ ssi pour tout contexte C,

$$C[M] \downarrow \quad \Leftrightarrow \quad C[N] \downarrow .$$

3. Proposez une sémantique dénotationelle de PCF finitaire. Montrez que votre sémantique est correcte.
4. On appelle λ terme un terme de PCF qui ne contient pas $\bot$. On note $M \Rightarrow N$ la relation $\to$ étendue à tout contexte (et pas seulement aux contextes d'évaluation). On admet le résultat suivant : pour tout λ terme M (y compris avec des variables libres), il existe un terme N tel que $M \Rightarrow^* N \not\Rightarrow$.
   Déduisez-en que votre sémantique est adéquate. Quelle est la sémantique de $\lambda x : \mathbf{bool}.\bot$?

Exercice 3. On considère le langage IMP défini par la grammaire

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

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

$$t ::= e \mid i \qquad \text{(termes logiques)}$$
$$\varphi ::= t = t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists i.\varphi \qquad \text{(formules logiques)}$$

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 :

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

   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 $[\![c]\!] : \mathit{Env} \to \mathit{Env}_\bot$ vue en en cours :

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

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

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

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,

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

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 $\mathbf{while}\ z < 3n\ \mathbf{do}\ z := z + 3\ ;\ y := y + 2z - 3\ ;\ x := x + y - z + 1$.

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 ?