TD 7

Exercice 1. 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) Etendez 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 ?

$$\frac{}{\rho \vdash x := e \Rightarrow \rho[x \mapsto [\![e]\!]\rho]}\ (:=) \qquad \frac{}{\rho \vdash \texttt{skip} \Rightarrow \rho}\ (\texttt{Skip})$$

$$\frac{\rho \vdash c_1 \Rightarrow \rho' \quad \rho' \vdash c_2 \Rightarrow \rho''}{\rho \vdash c_1; c_2 \Rightarrow \rho''}\ (\texttt{Seq})$$

$$\frac{\rho \vdash c_1 \Rightarrow \rho'}{\rho \vdash \texttt{if } e \texttt{ then } c_1 \texttt{ else } c_2 \Rightarrow \rho'}\ (\texttt{if}_1) \text{ si } [\![e]\!]\rho \neq 0 \qquad \frac{\rho \vdash c_2 \Rightarrow \rho''}{\rho \vdash \texttt{if } e \texttt{ then } c_1 \texttt{ else } c_2 \Rightarrow \rho'}\ (\texttt{if}_2) \text{ si } [\![e]\!]\rho = 0$$

$$\frac{\rho \vdash c \Rightarrow \rho' \quad \rho' \vdash \texttt{while } e \texttt{ do } c \Rightarrow \rho''}{\rho \vdash \texttt{while } e \texttt{ do } c \Rightarrow \rho''}\ (\texttt{while}) \text{ si } [\![e]\!]\rho \neq 0 \qquad \frac{}{\rho \vdash \texttt{while } e \texttt{ do } c \Rightarrow \rho}\ (\texttt{while}_{fin}) \text{ si } [\![e]\!]\rho = 0$$

Figure 1: La sémantique opérationnelle à grands pas de IMP

2. Expressions arithmétiques avec effets de bords: on étend nos expressions arithmétiques par la syntaxe suivante:

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

   Intuitivement, pour évaluer l'expression c `resultis` e, on évalue d'abord la commande c, puis on évalue e dans l'environnement obtenu.

   (a) Donnez des sémantiques opérationnelle et dénotationnelle pour ces expressions (on supposera disposer d'une sémantique pour les programmes c).

(b) Comment s'évalue le terme $((x := x + 1)$ `resultis` $x) + ((x := x + x)$ `resultis` $y)$ dans l'environnement $\rho = [x \mapsto 3]$.

(c) Vous comprenez désormais les horreurs que permet d'écrire le C: $T[i++] \leftarrow i++$.

Exercice 2. Une entrevue avec le π-calcul:
Le π-calcul est un langage théorique utilisé pour décrire des problématiques de concurrence. On suppose disposer d'un ensemble dénombrable de variables $\mathcal{X}$, un ensemble d'expressions $\mathcal{E}$ contenant l'ensemble de variables, et un ensemble fini de canaux de communication $\mathcal{C}$. On s'intéresse à un fragment du π-calcul dont la syntaxe est donnée par la grammaire suivante:

$$P ::= \texttt{skip} \mid in(c,x) \cdot P \mid out(c,e) \cdot P \mid P|Q$$

où $c \in \mathcal{C}$, $x \in \mathcal{X}$ et $e \in \mathcal{E}$.

Nous gardons ici l'ensemble des expressions parfaitement abstrait, mais on pourrait imaginer que $\mathcal{E}$ est l'ensemble des expressions arithmétiques, comme rencontré jusque là.

$$\frac{}{in(c,x) \cdot P \xRightarrow{in(c,x)} P}\ (in) \qquad \frac{}{out(c,e) \cdot P \xRightarrow{out(c,e)} P}\ (out)$$

$$\frac{P \overset{\alpha}{\Rightarrow} P'}{P|Q \overset{\alpha}{\Rightarrow} P'|Q}\ (left-par) \qquad \frac{Q \overset{\alpha}{\Rightarrow} Q'}{P|Q \overset{\alpha}{\Rightarrow} P|Q'}\ (right-par)$$
$$\text{pour } \alpha \in \{\tau, in(c,x), out(c,x) \mid c \in \mathcal{C}, x \in \mathcal{X}\}$$

$$\frac{P \xRightarrow{in(c,x)} P' \quad Q \xRightarrow{out(c,e)} Q'}{P|Q \overset{\tau}{\Rightarrow} P'[x \leftarrow e]|Q'}\ (left-sim-par)$$

$$\frac{P \xRightarrow{out(c,e)} P' \quad Q \xRightarrow{in(c,x)} Q'}{P|Q \overset{\tau}{\Rightarrow} P'|Q'[x \leftarrow e]}\ (right-sim-par)$$

Figure 2: Une sémantique du π-calcul

L'objectif de cet exercice est de montrer l'équivalence de deux sémantiques du π-calcul. La première est présentée en figure 2. La seconde est donnée ci dessous. Un multi-ensemble d'éléments de X est intuitivement un ensemble dans lequel un même élément peut intervenir plusieurs fois. Formellement, cela se représente comme une fonction de X dans $\mathbb{N}$, où à chaque élement $x \in X$ est associé le nombre de copies de x présentes dans le multi-ensemble. Nous travaillons avec des multi-ensembles finis, c'est à dire que l'ensemble des éléments présents dans le multi-ensemle est fini. Une autre façon de se le représenter: un multi-ensemble est une liste dans laquelle l'ordre des éléments n'importe pas. Nous les noterons comme des ensembles. Ainsi, les multi-ensembles $\{2; 2; 1\}$ et $\{2; 1; 2\}$ sont égaux, mais sont différents du multi-ensemble $\{2; 1\}$.

Etant donnés deux multi-ensembles S et T, l'union disjointe, noté $S+T$ correspond au multi-ensemble M tel que pour tout $x \in X$, $M(x) = S(x) + T(x)$. Les règles de la seconde sémantique du π-calcul sont les suivantes:

$$\{P|Q\} + S \to \{P; Q\} + S$$
$$\{in(c,x) \cdot P; out(c,e) \cdot Q\} + S \to \{P[x \leftarrow e]; Q\} + S$$

1. Enoncer et prouver la propriété d'équivalence des deux sémantiques.

Exercice 3. Soit c un programme et ρ un environnement. Montrez que s'il existe une suite infinie $(c \cdot \varepsilon, \rho) = q_0 \to q_1 \to \cdots \to q_n \to \ldots$, alors pour tout environnement ρ_∞, le jugement $\rho \vdash c \Rightarrow \rho_\infty$ n'est pas dérivable.

Exercice 4. Treillis complet:

Montrer que l'ensemble des parties d'un ensemble A quelconque est un treillis complet.

Theorem 1 (Knaster-Tarski)**.** Soit $(X, \leq)$ un treillis complet et $f : X \to X$ une fonction monotone. Alors f a un plus petit point fixe $\mathrm{lfp}(f)$ et un plus grand point fixe $\mathrm{gfp}(f)$. De plus, l'ensemble $Fix(f)$ des points fixes de f est un treillis complet (pour $\leq$).

Exercice 5. Sur le théorème de Knaster-Tarski:

1. Finissez la démonstration vue en cours.
2. Démontrez le thérorème de Cantor-Bernstein: si A et B sont deux ensembles tels qu'il existe une injection f de A dans B et une injection g de B dans A, alors A et B sont équipotents.

   Indication: chercher $X \subseteq A$ tel que pour $Y = B \setminus f(X)$, on a $g(Y) = A \setminus X$.
3. Montrez qu'une fonction $f : [0,1] \to [0,1]$ croissante admet un point fixe.
4. Montrez que si $f : [0,1] \to [0,1]$ est une fonction croissante alors:

   $f(\bigvee_{n\in\mathbb{N}} x_n) = \bigvee_{n\in\mathbb{N}} f(x_n)$ pour toute suite croissante à valeurs dans $[0,1]$ ssi f est continue à gauche sur $[0,1]$
5. En déduire que $\bigvee_{n\in\mathbb{N}} f^n(\bot)$ n'est pas toujours un point fixe de f.

Exercice 6. Dessinez les ensembles ordonnés suivants et indiquez lesquels sont des DCPO. Justifiez.

1. $\mathbf{Bool}_\bot = \{0, 1, \bot\}$ avec $x < y$ ssi $x = \bot$ et $y \neq \bot$.
2. $\mathbb{N}$ avec l'ordre usuel (*i.e.* ω pour les intimes).
3. $\mathbb{N}_\bot$ avec $x < y$ ssi $x = \bot$ et $y \in \mathbb{N}$.
4. $\mathbb{N} \cup \{\infty\}$ avec l'ordre usuel étendu par $n \leq \infty$ pour tout n (*i.e.* $\omega + 1$ pour les intimes).
5. $\{[x, y] \mid x, y \in I, x \leq y\}$ avec l'odre $\supseteq$ ($I = [0,1] \subseteq \mathbb{R}$).
6. $\{[x, y] \mid x, y \in I \cap \mathbb{Q}, x \leq y\}$ avec l'odre $\supseteq$.