TD 10

$$\llbracket x_\tau \rrbracket_\rho = \rho(x_\tau) \qquad \llbracket \dot{n} \rrbracket_\rho = n \qquad \llbracket uv \rrbracket_\rho = \llbracket u \rrbracket_\rho(\llbracket v \rrbracket_\rho)$$

$$\llbracket \texttt{letrec}\ f_{\sigma\to\tau}(x_\sigma) = u\ \texttt{in}\ v \rrbracket_\rho = \llbracket v \rrbracket(\rho[f_{\sigma\to\tau} \mapsto \mathrm{lfp}(F^\rho_{f_{\sigma\to\tau},x_\sigma,u})])$$

$$\text{où } F^\rho_{f_{\sigma\to\tau},x_\sigma,u}(\phi) = (V \in \llbracket\sigma\rrbracket \mapsto \llbracket u \rrbracket(\rho[f_{\sigma\to\tau} \mapsto \phi, x_\sigma \mapsto V]))$$

$$\llbracket u \dot{+} v \rrbracket_\rho = \begin{cases} \llbracket u \rrbracket_\rho + \llbracket v \rrbracket_\rho & \text{si } \llbracket u \rrbracket_\rho \neq \bot,\ \llbracket v \rrbracket_\rho \neq \bot \\ \bot & \text{sinon} \end{cases}$$

$$\llbracket \dot{-} u \rrbracket_\rho = \begin{cases} -\llbracket u \rrbracket_\rho & \text{si } \llbracket u \rrbracket_\rho \neq \bot \\ \bot & \text{sinon} \end{cases}$$

$$\llbracket \textbf{if } u = 0 \textbf{ then } v \textbf{ else } w \textbf{ fi} \rrbracket_\rho = \begin{cases} \llbracket v \rrbracket_\rho & \text{si } \llbracket u \rrbracket_\rho = 0 \\ \llbracket w \rrbracket_\rho & \text{si } \bot \neq \llbracket u \rrbracket_\rho \neq 0 \\ \bot & \text{sinon} \end{cases}$$

FIGURE 1 – Sémantique dénotationnelle de PCF en appel par nom

$$\frac{}{E \vdash x_\tau \Rightarrow E(x_\tau)} \qquad \frac{}{E \vdash \dot{n} \Rightarrow \dot{n}}$$

$$\frac{E \vdash u \Rightarrow \langle \texttt{rec} f_{\sigma\to\tau}(x_\sigma) = u', E' \rangle \qquad E \vdash v \Rightarrow p \qquad E'[f_{\sigma\to\tau} \mapsto \langle \texttt{rec} f_{\sigma\to\tau}(x_\sigma) = u', E' \rangle,\ x_\sigma \mapsto p] \vdash u' \Rightarrow p'}{E \vdash uv \Rightarrow p'}$$

$$\frac{E[f_{\sigma\to\tau} \mapsto \langle \texttt{rec} f_{\sigma\to\tau}(x_\sigma) = u, E \rangle] \vdash v \Rightarrow p}{E \vdash \texttt{letrec}\ f_{\sigma\to\tau}(x_\sigma) = u\ \texttt{in}\ v \Rightarrow p}$$

$$\frac{E \vdash u \Rightarrow p_1 \qquad E \vdash v \Rightarrow p_2}{E \vdash u \dot{+} v \Rightarrow p_1 + p_2} \qquad \frac{E \vdash u \Rightarrow p}{E \vdash \dot{-} u \Rightarrow -p}$$

$$\frac{E \vdash u \Rightarrow 0 \qquad E \vdash v \Rightarrow p}{E \vdash \textbf{if } u = 0 \textbf{ then } v \textbf{ else } w \textbf{ fi} \Rightarrow p} \qquad \frac{E \vdash u \Rightarrow n \neq 0 \qquad E \vdash w \Rightarrow p}{E \vdash \textbf{if } u = 0 \textbf{ then } v \textbf{ else } w \textbf{ fi} \Rightarrow p}$$

FIGURE 2 – Sémantique à grands pas de PCF_V

Exercice 1. On considère l'expression PCF u suivante :

$$\begin{array}{l} \texttt{letrec}\ f_{\texttt{int}\to\texttt{int}}(x_{\texttt{int}}) = \dot{3}\ \texttt{in} \\ \quad \texttt{letrec}\ g_{\texttt{int}\to\texttt{int}}(x_{\texttt{int}}) = g_{\texttt{int}\to\texttt{int}}(x_{\texttt{int}})\ \texttt{in} \\ \quad\quad f_{\texttt{int}\to\texttt{int}}(g_{\texttt{int}\to\texttt{int}}\dot{0}) \end{array}$$

Montrer que $\llbracket u \rrbracket\rho = 3$ pour n'importe quel environnement ρ. Montrer que pourtant, on ne peut pas dériver $E \vdash u \Rightarrow 3$ pour aucun P-environnement E : en fait, il n'y a pas de dérivation $E \vdash u \Rightarrow p$ pour aucun E et aucun p. En déduire que la sémantique opérationnelle de PCF_V n'est pas adéquate par rapport à la sémantique dénotationnelle de PCF en appel par nom.

$$
\begin{aligned}
[\![x_\tau]\!]^\circ_{\rho\kappa} &= \kappa(\rho(x_\tau)) \qquad\qquad [\![\dot{n}]\!]^\circ_{\rho\kappa} = \kappa(n)\\
[\![uv]\!]^\circ_{\rho\kappa} &= [\![u]\!]^\circ_\rho(f \in [\![\sigma \to \tau]\!]^\circ \mapsto [\![v]\!]^\circ_{\rho(f\kappa)})\\
[\![\texttt{letrec}\ f_{\sigma\to\tau}(x_\sigma) = u\ \texttt{in}\ v]\!]^\circ_{\rho\kappa} &= [\![v]\!]^\circ(\rho[f_{\sigma\to\tau} \mapsto \mathrm{lfp}(T^\rho_{f_{\sigma\to\tau},x_\sigma,u})])\kappa\\
&\quad \text{où } T^\rho_{f_{\sigma\to\tau},x_\sigma,u}(\phi) = (\kappa \in [\![\sigma]\!]^\perp \mapsto\\
&\qquad (V \in [\![\sigma]\!]^\circ \mapsto [\![u]\!]^\circ(\rho[f_{\sigma\to\tau} \mapsto \phi, x_\sigma \mapsto V])\kappa))\\
[\![u \dot{+} v]\!]^\circ_{\rho\kappa} &= [\![u]\!]^\circ_\rho(m \in \mathbb{Z} \mapsto [\![v]\!]^\circ_\rho(n \in \mathbb{Z} \mapsto \kappa(m+n)))\\
[\![\dot{-}u]\!]^\circ_{\rho\kappa} &= [\![u]\!]^\circ_\rho(n \in \mathbb{Z} \mapsto \kappa(-n))\\
[\![\textbf{if } u = 0 \textbf{ then } v \textbf{ else } w \textbf{ fi}]\!]^\circ_{\rho\kappa} &= [\![u]\!]\rho\left(n \in \mathbb{Z} \mapsto \begin{cases} [\![v]\!]_{\rho\kappa} & \text{si } n = 0\\ [\![w]\!]_{\rho\kappa} & \text{sinon}\end{cases}\right)
\end{aligned}
$$

FIGURE 3 – Sémantique dénotationnelle de PCF_V par passage à la continuation

Exercice 2. 1. Montrer que $[\![(y+x)+(-(1+y))]\!]^\circ_{[x\mapsto 3, y\mapsto 2]\mathrm{id}_\mathbb{Z}} = 2$, où $\mathrm{id}_\mathbb{Z} = (n \in \mathbb{Z} \mapsto n)$.

On se donne pour chaque type τ un ensemble $Addr_\tau$ d'adresses pointant vers des objets de type τ. On pose St le dcpo des *stores*, qui est l'ensemble des fonctions totales qui à chaque $a \in Addr_\tau$ associe un élément de $[\![\tau]\!]^\circ$, avec l'ordre usuel. ***Mini-Caml*** est le langage PCF_V plus deux constructeurs de type `unit` et `ref` et trois nouvelles constructions syntaxiques :

— pour chaque expression u de type `ref` τ, $!u$ est une expression de type τ ;

— pour toutes expressions u : `ref` τ et $v : \tau$, $u := v$ est une expression de type `unit` ;

— pour toute expression $u : \tau$, `ref`(u) est une expression de type `ref` τ

On étend la sémantique par continuation de PCV_V en posant $[\![\texttt{unit}]\!]^\circ = \{\perp, \top\}$ avec $\perp < \top$, $[\![\texttt{ref}\ \tau]\!]^\circ = Addr_\tau$, et surtout en posant $Ans = [Mem \to Ans_0]$ pour un noveau dcpo pointé Ans_0 de réponses dites pures. La sémantique des constructions déjà présentes en PCF_V reste définie par la Figure 2 et la sémantique des nouvelles constructions est :

$$
\begin{aligned}
[\![!u]\!]^\circ_{\rho\kappa} &= [\![u]\!]^\circ_\rho(\text{deref } \kappa)\\
[\![u := v]\!]^\circ_{\rho\kappa} &= [\![u]\!]^\circ_\rho(a \in Addr_\tau \mapsto [\![v]\!]^\circ_\rho(\text{assign } a\ k))\\
[\![\texttt{ref}(u)]\!]^\circ_{\rho\kappa} &= [\![u]\!]^\circ_\rho(\text{new } \kappa)
\end{aligned}
$$

où :

$$
\begin{aligned}
(\text{deref } \kappa)(a)(\mu) &= \kappa(\mu(a))(\mu)\\
(\text{assign } a\ \kappa)(V)(\mu) &= \kappa(\top)(\mu[a \mapsto V])\\
(\text{new } \kappa)(V)(\mu) &= \begin{cases} \kappa(a)(\mu[a \mapsto V]) & \text{si } a = alloc_\tau(\mu) \neq \perp\\ \perp & \text{sinon}\end{cases}
\end{aligned}
$$

où $alloc_\tau : St \to (Addr_\tau)_\perp$ est une fonction retournant ou bien $\perp$ ou bien une adresse de type τ hors du domain du store passé en argument. On suppose $alloc_\tau$ continue, en voyant $Addr_\tau$ comme un dcpo avec l'ordre d'égalité.

1. Écrire une sémantique à grands pas de ce nouveau langage PCV_V + St, dans le style de celle de PCF_V, mais dont les jugements auront la forme $E, S \vdash u \Rightarrow p, S'$. La nouveauté est la présence des S et S', qui sont des fonctions partielles associant à chaque adresse $a \in Addr_\tau$ une valeur de type τ.
2. Montrer la correction de la sémantique dénotationnelle de PCF_V + St par raport à cette sémantique opérationnelle.

Exercice 3. On considère le langage de programmation suivant.

$$\begin{array}{ll} e ::= x \mid \mathbf{null} & b ::= e = e' \mid \neg b \mid b \wedge b \qquad (n \in \mathbb{N}) \\ p ::= & \mathbf{skip} \mid x := e \mid x := *y \mid *x := e \mid x := \mathbf{ref}(n) \mid \mathbf{free}(x) \\ & p_1 ; p_2 \mid \mathbf{if}\ b\ \mathbf{then}\ p_1\ \mathbf{else}\ p_2\ \mathbf{fi} \end{array}$$

On modélise un état mémoire par un couple (s, h) tel que $s : \mathsf{Var} \to \mathbb{N}$ assigne à chaque variable une adresse, et $h : \mathbb{N} \to \mathbb{N}_\perp$ assigne à chaque addresse son contenu (lui-même une adresse), de sorte que $h(n) = \perp$ si l'adresse n n'est pas allouée. On prend par convention que null est l'adresse 0, et qu'elle n'est jamais allouée. On note Σ l'ensemble des états mémoires.

1. On suppose la procédure d'allocation mémoire non déterministe. Proposez une sémantique dénotationelle $[\![p]\!] : \Sigma \to \mathcal{P}(\Sigma)^\top$ qui à tout état (s, h) associe l'ensemble des états (s', h') possibles à la fin de l'exécution de p si p ne fait pas d'erreur quelque soit les choix non déterministes, et $\top$ sinon.
2. Pourquoi la règle suivante, valide en logique de Hoare classique, cesse-t-elle d'être valide pour notre langage à pointeurs ?

$$\frac{\{\varphi \wedge \varphi_F\}\ p\ \{\psi \wedge \varphi_F\} \qquad \mathsf{modvars}(p) \cap \mathsf{vars}(\varphi) = \emptyset}{\{\varphi\}\ p\ \{\psi\}}$$

où $\varphi ::= t = t \mid \neg\varphi \mid \varphi \wedge \varphi$
$t ::= n(\in \mathbb{N}) \mid x \mid *x \mid \ldots$

Exercice 4. 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 triples 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.

$$\frac{(\varphi, \psi).\Pi \vdash \{\varphi \wedge b\}\ c; \mathbf{continue}\ \{\perp\} \qquad \varphi \wedge \neg b \models \psi}{\Pi \vdash \{\varphi\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{\psi\}}$$

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

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

2. On appelle continuation une fonction $\kappa : \mathbb{Z}^{\mathsf{Var}} \to (\mathbb{Z}^{\mathsf{Var}})_\perp$ 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 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*, $\mathsf{safe}(k, \varphi)$, si pour tout environnement ρ, $\rho \models \varphi$ implique $k(\rho) = \perp$.
   — Une pile de paires de continuation $\pi = (k_{c,1}, k_{b,1}) \ldots (k_{c,n}, k_{b,n})$ est sûre pour une pile de paire de formules $\Pi = (\varphi_1, \psi_1) \ldots (\varphi_n, \psi_n)$, $\mathsf{safe}(\pi, \Pi)$, si pour tout $i = 1 \ldots n$, $\mathsf{safe}(k_{c,i}, \varphi_i)$ 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.

Exercice 5. On utilise les sémantiques dénotationnelle et opérationnelle en appel par nom de PCF.

Un terme M est clos si et seulement si il n'a aucune variable libre, i.e. $\mathtt{fv}(M) = \emptyset$. On définit une famille de relations binaires $(R_\tau)_{\tau \text{ type}}$ telles que

1. pour chaque terme clos $M : \texttt{int}$ et $a \in \mathbb{Z}_\perp$,

$$MR_{\texttt{int}}a \text{ si et seulement si } (a = \perp \text{ ou } \emptyset \vdash M \Rightarrow a)$$

2. pour chaque terme $M : \sigma \to \tau$ et $f \in [\sigma \to \tau]$,

$$MR_{\sigma\to\tau}f \text{ si et seulement si pour tous } N, a \text{ si } NR_\sigma a \text{ alors } (MN)R_\tau f(a)$$

Montrer que :

1. pour chaque type τ et terme $M : \tau$, $MR_\tau = \{a \mid MR_\tau a\}$ est un fermé de Scott et contient $\perp$.
2. Pour tout type τ et tout termes $M, N : \tau$, si $M \Rightarrow N$ alors $NR_\tau \subseteq MR_\tau$.

Soit $\theta = [x_1 \mapsto M_1 : \tau_1, \ldots, x_n \mapsto M_n : \tau_n]$ une substitution des variables vers les termes. θ est close si et seulement si pour tout $i \in [\![1, n]\!]$ M_i est clos. Soit ρ un environnement. On va écrire $\theta R_\star \rho$ pour $\forall i \in [\![1, n]\!]$, $M_i R_{\tau_i} \rho(x_i)$. Montrer que :

3. Pour chaque substitution close θ, pour chaque environnement ρ et pour chaque terme $M : \tau$, si $\texttt{fv}(M) \subseteq \texttt{dom}(\theta)$ et $\theta R_\star \rho$ alors $(M\theta)R_\tau [\![M]\!]_\rho$.
4. En deduire que si M est une expression de type $\texttt{int}$ sans variable libre et si $[\![M]\!]_\emptyset = n \neq \perp$, alors $u \Rightarrow \dot{n}$ est dérivable.