DM Programmation I (2016-17)

Nous allons revenir au langage jouet Imp du cours :

Commandes				Expressions			
c	::=	$x := e$	affectation	e	::=	x	variables
	|	`skip`	ne rien faire		|	$\dot{n}$	constante entière ($n \in \mathbb{Z}$)
	|	$c_1; c_2$	séquence		|	$e \dot{+} e$	addition
	|	`if` e `then` c_1 `else` c_2	conditionnelle		|	$\dot{-} e$	opposé
	|	`while` e `do` c	boucle while				

Les variables sont supposées en nombre fini, et seront numérotées x_1, x_2, …, x_n.

On cherche à déterminer des intervalles de variation possibles des variables à travers l'exécution d'un programme. Pour ceci, on va considérer une sémantique dénotationnelle « quotient ».

On note $\mathcal{I}$ l'ensemble contenant l'élément spécial $\bot$, plus tous les couples (a, b), où $a, b \in \mathbb{Z} \cup \{-\infty, +\infty\}$, et $a + 1 < b$. On peut penser au couple (a, b) comme à une notation représentant l'intervalle d'entiers $]a, b[$ $(= \{c \in \mathbb{Z} \mid a < c < b\})$, et à $\bot$ comme à un symbole qui signifie l'intervalle vide.

On ordonne $\mathcal{I}$ par $v \leq w$ ssi :

— $v = \bot$,

— ou $v = (a, b)$, $w = (c, d)$, et $a \geq c$, $b \leq d$.

Ce n'est pas un ordre total ! Tout raisonnement de la forme « montrons que $u \leq v$; si on avait $u > v$, alors … contradiction » est donc faux. Dans la suite, ne confondez pas non plus « plus grand élément » (le plus grand élément de A s'il existe un est $v \in A$ tel que pour tout $u \in A$, $u \leq v$) et « élément maximal » (un élément maximal de A est un $v \in A$ tel que pour tout $u > v$, u n'est pas dans A). Ne confondez pas non plus ces notions avec « borne supérieure » : la borne supérieure de A, si elle existe, est le plus petit des majorants. Un majorant de A est un élément v, pas nécessairement dans A, tel que tout $u \in A$ est $\leq v$. La borne supérieure w de A, si elle existe, est le plus petit élément de l'ensemble des majorants : w est un majorant de A, et pour tout majorant v de A, $w \leq v$.

Définissons l'addition dans $\mathcal{I}$ par :

— $\bot + v = v + \bot = \bot$ pour tout $v \in \mathcal{I}$;

— $(a, b) + (c, d) = (a + c, b + d)$ (on conviendra que $(-\infty) + c = -\infty$ et $b + (+\infty) = (+\infty)$; noter que l'on n'aura jamais à calculer l'expression absurde $(-\infty) + (+\infty)$, au vu des contraintes de formation des éléments de $\mathcal{I}$).

> *Note : il y a une erreur ici ; je voulais que + représente l'addition, auquel cas j'aurais dû écrire* $(a, b) + (c, d) = (a + c + 1, b + d - 1)$ *; ça ne prête cependant pas à conséquence dans le reste du sujet.*

Similairement, définissons l'opposé dans $\mathcal{I}$ par : $-\bot = \bot$, $-(a,b) = (-b,-a)$ sinon.

1. $\mathcal{I}$ est un treillis complet, autrement dit toute famille d'éléments a une borne supérieure et une borne inférieure, comme il est facile de le voir. Comment est définie l'opération $\vee$, borne supérieure de deux éléments ? (Oui, $v \vee w$ est le plus petit des majorants de $\{v,w\}$. Je veux une définition explicite, avec analyse des différents cas possibles.)

   $\bot \vee v = v$, $v \vee \bot = v$, *et sinon* $(a,b) \vee (c,d) = (\min(a,c), \max(b,d))$.

2. $\mathcal{I}^n$, l'ensemble des n-uplets d'éléments de $\mathcal{I}$, avec l'ordre composante par composante, est-il : (a) un dcpo ? (b) un treillis complet ?

   (a) et (b). C'est un treillis complet, donc un dcpo. Pour montrer que c'est un treillis complet, il suffit de vérifier que tout produit de treillis complets est un treillis complet, le sup dans le produit étant calculé composante par composante.

On identifiera $\mathcal{I}^n$ à l'espace des fonctions de l'ensemble des variables vers $\mathcal{I}$, c'est-à-dire aux environnements « quotients ». Pour $\eta \in \mathcal{I}^n$, $\eta(x_i)$ sera donc la $i$ème composante du n-uplet η.

On étend la notation $\vee$ aux environnements par : $\eta \vee \eta'$ envoie toute variable x vers $\eta(x) \vee \eta'(x)$.

3. Une fonction monotone $F : \mathcal{I}^n \to \mathcal{I}^n$ est dite *inflationnaire* si et seulement si $\eta \leq F(\eta)$ pour tout η. Pour toute fonction inflationnaire F, pour tout $\eta_0 \in \mathcal{I}^n$, montrer que F a plus petit point fixe $\geq \eta_0$. A titre d'indication, considérez la fonction F' définie par $F'(\eta) = \eta_0 \vee F(\eta)$.

   On notera dans la suite $\mathrm{lfp}_\eta(F)$ ce plus petit point fixe de F au-dessus de η.

   F' est monotone, comme composée de fonctions monotones. Elle a donc a un plus petit point fixe par le théorème de Tarski. Appelons-le η.
   $F'(\eta) = \eta$, donc $\eta_0 \vee F(\eta) = \eta$ donc $F(\eta) \leq \eta$ et $\eta_0 \leq \eta$.
   Comme F est inflationnaire, $\eta \leq F(\eta)$, et avec $F(\eta) \leq \eta$ ceci implique $\eta = F(\eta)$. Donc η est un point fixe de F, et il est plus grand ou égal à η_0.
   Si η' est un autre point fixe de F au-dessus de η_0, alors $F'(\eta') = \eta_0 \vee \eta' = \eta'$, donc η' est un point fixe de F', et est donc au-dessus de η. Donc η est le plus petit point fixe de F au-dessus de η_0.

La sémantique quotient $Q[\![e]\!]$ des expressions e prend un environnement quotient η, et retourne une valeur dans $\mathcal{I}$, selon les clauses :

$$\begin{aligned} Q[\![x]\!]\eta &= \eta(x) \\ Q[\![\dot{n}]\!]\eta &= (n-1, n+1) \\ Q[\![e_1 \dot{+} e_2]\!]\eta &= Q[\![e_1]\!]\eta + Q[\![e_2]\!]\eta \\ Q[\![\dot{-}e]\!]\eta &= -Q[\![e]\!]\eta \end{aligned}$$

La sémantique quotient $Q[\![c]\!]$ des commandes c prend un environnement quotient η, et

retourne un nouvel environnement quotient, selon les règles :

$$
\begin{aligned}
Q[\![x := e]\!]\eta &= \eta \vee \eta[x \mapsto Q[\![e]\!]\eta]) && (1)\\
Q[\![\texttt{skip}]\!]\eta &= \eta && (2)\\
Q[\![c_1; c_2]\!]\eta &= Q[\![c_2]\!](Q[\![c_1]\!]\eta) && (3)\\
Q[\![\texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2]\!]\eta &= \begin{cases} \eta & \text{si } Q[\![e]\!]\eta = \bot \\ Q[\![c_2]\!]\eta & \text{si } Q[\![e]\!]\eta = (-1,1) \\ Q[\![c_1]\!]\eta & \text{si } Q[\![e]\!]\eta = (a,b) \text{ avec } a \geq 0 \text{ ou } b \leq 0 \\ Q[\![c_1]\!]\eta \vee Q[\![c_2]\!]\eta & \text{sinon} \end{cases} && (4)\\
Q[\![\texttt{while}\ e\ \texttt{do}\ c]\!]\eta &= \mathrm{lfp}_\eta(F_{e,c}) && (5)
\end{aligned}
$$

où $F_{e,c} \colon \mathcal{I}^n \to \mathcal{I}^n$ est la fonction suivante :

$$
F_{e,c}(\eta') = \begin{cases} \eta' & \text{si } Q[\![e]\!]\eta' = \bot \text{ ou } Q[\![e]\!]\eta' = (-1,1) \\ \eta' \vee Q[\![c]\!]\eta' & \text{sinon.} \end{cases}
$$

On souhaite montrer que cette définition est sensée.

4. En supposant que $Q[\![e]\!]$ et $Q[\![c]\!]$ sont déjà définies et monotones en leur argument η, montrez que $F_{e,c}$ est inflationnaire de $\mathcal{I}^n$ vers $\mathcal{I}^n$. N'oubliez pas de démontrer la monotonie d'abord.

   On montre d'abord la monotonie. Le point important est que sur les deux cas définissant $F_{e,c}(\eta')$, lorsque η' augmente, on peut soit rester dans le même cas, soit passer du premier au deuxième, mais jamais l'inverse.

   Formellement, supposons $\eta \leq \eta'$. Si $Q[\![e]\!]\eta$ vaut $\bot$ ou $(-1,1)$ et $Q[\![e]\!]\eta'$ vaut aussi $\bot$ ou $(-1,1)$, alors $F_{e,c}(\eta) = \eta \leq \eta' = F_{e,c}(\eta')$. Si $Q[\![e]\!]\eta$ vaut $\bot$ ou $(-1,1)$ mais que ce n'est pas le cas de $Q[\![e]\!]\eta'$, alors $F_{e,c}(\eta) = \eta \leq \eta' \leq \eta' \vee Q[\![c]\!]\eta' = F_{e,c}(\eta')$. Finalement, si $Q[\![e]\!]\eta$ ne vaut ni $\bot$ ni $(-1,1)$, alors c'est le cas aussi de $Q[\![e]\!]\eta'$, car $Q[\![e]\!]\eta \leq Q[\![e]\!]\eta'$ et les seuls éléments $\leq \bot$ ou $\leq (-1,1)$ sont $\bot$ et $(-1,1)$. Donc $F_{e,c}(\eta) = \eta \vee Q[\![c]\!]\eta \leq \eta' \vee Q[\![c]\!]\eta' = F_{e,c}(\eta')$, où l'égalité du milieu est justifiée par le fait que $\vee$ est monotone et que $Q[\![c]\!]$ est monotone.

   Pour l'inflationnarité proprement dite, quel que soit le cas dans lequel on se place, $\eta \leq F_{e,c}(\eta)$, de façon claire.

5. La question 3 permet donc d'en conclure que $Q[\![\texttt{while}\ e\ \texttt{do}\ c]\!]$ est bien définie. Pourquoi ceci définit-il bien une fonction monotone de $\mathcal{I}^n$ dans $\mathcal{I}^n$? Autrement dit, supposons $\eta \leq \eta'$, alors pourquoi a-t-on $\mathrm{lfp}_\eta(F_{e,c}) \leq \mathrm{lfp}_{\eta'}(F_{e,c})$? On montrera, plus généralement, que si F est une fonction inflationnaire d'un treillis complet L dans lui-même (par exemple $F_{e,c}$, mais pas uniquement), et si $\eta \leq \eta'$, alors $\mathrm{lfp}_\eta(F) \leq \mathrm{lfp}_{\eta'}(F)$.

   Si $\eta \leq \eta'$, alors le plus petit point fixe de F au-dessus de η' est aussi un plus petit point fixe de F au-dessus de η, donc est au-dessus du plus petit point fixe de F au-dessus de η.

A partir de ces considérations, on peut démontrer que $Q[\![c]\!]$ est bien définie pour toute commande c. On peut aussi démontrer que c'est une fonction Scott-continue de $\mathcal{I}^n$ vers $\mathcal{I}^n$. Parmi ce qu'il faut démontrer dans ce but, on trouver les trois questions suivantes.

6. Montrer que la fonction $+ : \mathcal{I} \times \mathcal{I} \to \mathcal{I}$ est Scott-continue. En clair, vous devrez montrer que : (a) + est monotone, (b) pour toute famille dirigée $(v_i, w_i)_{i \in I}$ de couples d'éléments de $\mathcal{I}$, $\sup_{i \in I}(v_i + w_i) = (\sup_{i \in I} v_i) + (\sup_{i \in I} w_i)$. On pourra utiliser sans démonstration les résultats suivants caractérisant les bornes supérieures de familles dirigées $(v_i)_{i \in I}$ dans $\mathcal{I}$: si tous les v_i sont égaux à $\bot$, alors $\sup_{i \in I} v_i = \bot$; sinon, écrivons v_i sous la forme (a_i, b_i) pour tout $i \in I$ tel que $v_i \neq \bot$, alors $\sup_{i \in I} v_i = (\inf_{i \in I, v_i \neq \bot} a_i, \sup_{i \in I, v_i \neq \bot} b_i)$, où les infs et les sups dans la dernière expression sont pris dans $\mathbb{Z} \cup \{-\infty, +\infty\}$ muni de son ordre usuel ($-\infty \leq \cdots \leq -3 \leq -2 \leq -1 \leq 0 \leq 1 \leq 2 \leq 3 \leq \cdots \leq +\infty$).

   (a) D'abord, si $(a, b) \leq (a', b')$ et $(c, d) \leq (c', d')$, alors $a \geq a'$, $c \geq c'$, $b \leq b'$, $d \leq d'$, donc $a + c \geq a' + c'$ et $b + d \leq b' + d'$, c'est-à-dire $(a + c, b + d) \leq (a' + c', b' + d')$. Ensuite, si l'un des deux sommandes vaut $\bot$, disons $\bot \leq v$ et $w \leq w'$, alors $\bot + w = \bot$ est $\leq v + w'$, puisqu'inférieur ou égal à tout élément.

   (b) Soit (v, w) le sup de $(v_i, w_i)_{i \in I}$. Si tous les v_i sont égaux à $\bot$, alors $v = \bot$, et on a $v + w = \bot$, $\sup_{i \in I}(v_i + w_i) = \sup_{i \in I}(\bot + w_i) = \sup_{i \in I} \bot = \bot$. Pareil si tous les w_i sont égaux à $\bot$. Sinon, soit $J = \{i \in I \mid v_i \neq \bot\}$, $K = \{i \in I \mid w_i \neq \bot\}$. Pour tout $i \in J$, posons $v_i = (a_i, b_i)$, et pour tout $i \in K$, posons $w_i = (c_i, d_i)$.

   On s'aperçoit de : $(*)$ pour tout $j \in J$ et pour tout $k \in K$, il existe un $i \in J \cap K$ tel que $(v_j, w_j), (v_k, w_k) \leq (v_i, w_i)$. Il suffit pour cela d'invoquer le fait que I est dirigée, et que, puisque $v_i \geq v_j$ et $v_j \neq \bot$ (car $j \in J$), on a $v_i \neq \bot$ donc $i \in J$, et puisque $w_i \geq w_k$ et $w_k \neq \bot$ (car $k \in K$), on a $w_i \neq \bot$ donc $i \in K$.

   Ceci permet de montrer que $v = \sup_{i \in J} v_i = \sup_{i \in J \cap K} v_i$, et de même $w = \sup_{i \in K} w_i = \sup_{i \in J \cap K} w_i$. On a $v \geq \sup_{i \in J \cap K} v_i$ trivialement, et pour montrer l'inégalité réciproque, il suffit d'observer que pour tout $i \in J$, il existe un $i' \in J \cap K$ tel que $v_i \leq v_{i'}$. On utilise pour cela le fait que K est non vide (prenons $k \in K$) et la remarque $(*)$.

   De plus, la famille $J \cap K$ est dirigée. Elle est non vide, car J et K sont non vides, et l'on peut alors appliquer $(*)$ à un $j \in K$ et un $k \in K$. Si i_1 et i_2 sont deux éléments de $J \cap K$, alors comme I est dirigée, il existe un i dans I tel que $(v_{i_1}, w_{i_1}), (v_{i_2}, w_{i_2}) \leq (v_i, w_i)$. En particulier v_i est supérieur ou égal à v_{i_1} donc différent de $\bot$ (car $i_1 \in J$) donc $i \in J$, et de même $i \in K$.

   On a maintenant : $v + w = \sup_{i \in J \cap K} v_i + \sup_{i \in J \cap K} w_i = (\inf_{i \in J \cap K} a_i, \sup_{i \in J \cap K} b_i) + (\inf_{i \in J \cap K} c_i, \sup_{i \in J \cap K} d_i) = (\inf_{i \in J \cap K} a_i + \inf_{i \in J \cap K} c_i, \sup_{i \in J \cap K} b_i + \inf_{i \in J \cap K} d_i)$.

   On doit ensuite montrer que $+$ commute aux sups dirigés et aux infs filtrants (c'est-à-dire dirigés dans l'ordre inverse).

   On obtient ainsi $v + w = (\inf_{i \in J \cap K}(a_i + c_i), \sup_{i \in J \cap K}(b_i + d_i))$. Ceci est par définition inférieur ou égal à $\sup_{i \in I, v_i + w_i \neq \bot}(a_i + c_i, b_i + d_i)$, puisque pour tout $i \in J \cap K$, $(a_i + c_i, b_i + d_i) \neq \bot$, c'est-à-dire à $\sup_{i \in I}(v_i + w_i)$.

7. Montrer que la fonction $- : \mathcal{I} \to \mathcal{I}$ est Scott-continue.

   C'est beaucoup plus simple. D'abord, $-$ est monotone (malgré les apparences !) : si $v \leq w$, soit $v = \bot$ et $-v = \bot \leq -w$, soit $v = (a, b)$, $w = (c, d)$ avec $a \geq c$ et $b \leq d$, donc $-b \geq -d$ et $-a \leq -c$, c'est-à-dire $-v = (-b, -a) \leq (-d, -c) = -w$.

   Ensuite, si $(v_i)_{i \in I}$ est une famille dirigée de $\mathcal{I}$ de sup v, soit tous les v_i sont égaux à $\bot$, $v = \bot$, et alors tous les $-v_i$ sont égaux à $\bot$, donc $\sup_{i \in I}(-v_i) = \bot = -v$; soit l'ensemble J des i tels que $-v_i \neq \bot$ est le même que celui des i tels que $v_i \neq \bot$, donc $\sup_{i \in I}(-v_i) = (\inf_{i \in J} -b_i, \sup_{i \in J} -a_i) = (-\sup_{i \in J} b_i, -\inf_{i \in J} a_i) = -v$.

$$\frac{}{(x := e, \rho) \to \rho[x \mapsto [\![e]\!]\rho]}\ (\to :=) \qquad \frac{}{(\texttt{skip}, \rho) \to \rho}\ (\to \texttt{skip})$$

$$\frac{(c_1, \rho) \to \rho'}{(c_1; c_2, \rho) \to (c_2, \rho')}\ (\to Seq_{fin}) \qquad \frac{(c_1, \rho) \to (c_1', \rho')}{(c_1; c_2, \rho) \to (c_1'; c_2, \rho')}\ (\to Seq)$$

$$\frac{}{(\texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2, \rho) \to (c_1, \rho)}\ (\to \texttt{if}\ 1) \text{ si } [\![e]\!]\rho \neq 0 \qquad \frac{}{(\texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2, \rho) \to (c_2, \rho)}\ (\to \texttt{if}\ 2) \text{ si } [\![e]\!]\rho = 0$$

$$\frac{}{(\texttt{while}\ e\ \texttt{do}\ c, \rho) \to (c; \texttt{while}\ e\ \texttt{do}\ c, \rho)}\ (\to \texttt{while}) \text{ si } [\![e]\!]\rho \neq 0 \qquad \frac{}{(\texttt{while}\ e\ \texttt{do}\ c, \rho) \to \rho}\ (\to \texttt{while}_{fin}) \text{ si } [\![e]\!]\rho = 0$$

FIGURE 1 – Une sémantique opérationnelle à petits pas de IMP

8. Montrer que, si F est une fonction inflationnaire et Scott-continue de $\mathcal{I}$ dans $\mathcal{I}$, alors la fonction qui à $\eta \in \mathcal{I}$ associe $\mathrm{lfp}_\eta(F)$ est encore Scott-continue.

   $\mathrm{lfp}_\eta(F)$ *est égal à* $\mathrm{lfp}(F'_\eta)$*, où* $F'_\eta(\eta') = \eta \vee F(\eta')$ *(voir question 3).* F'_η *est Scott-continue parce que* F *et* $\vee$ *le sont. On peut ensuite appliquer le théorème vu en cours selon lequel* lfp *est une fonctionnelle Scott-continue, et réaliser que* F'_η *est elle-même Scott-continue en fonction de* η *(parce que* $\vee$ *est Scott-continue). Sinon, pour le démontrer à la main, on exploite le fait que* $\mathrm{lfp}(F'_\eta)$ *s'écrit comme* $\sup_{n\in\mathbb{N}} {F'_\eta}^n(\bot)$*, et que* $\sup_\eta$ *et* $\sup_{n\in\mathbb{N}}$ *commutent.*

   Une démonstration plus élémentaire est la suivante. On a déjà vu que lfp_η *est monotone en* η *à la question 5. Pour toute famille dirigée* $(\eta_i)_{i\in I}$ *de sup* η*, il reste à montrer* $\sup_{i\in I} \mathrm{lfp}_{\eta_i}(F) \geq \mathrm{lfp}_\eta(F)$*, l'inégalité réciproque étant une conséquence de la monotonie. Or* $F(\sup_{i\in I} \mathrm{lfp}_{\eta_i}(F)) = \sup_{i\in I} F(\mathrm{lfp}_{\eta_i}(F)) = \sup_{i\in I} \mathrm{lfp}_{\eta_i}(F)$ *en utilisant le fait que* F *est Scott-continue puis que chaque* $\mathrm{lfp}_{\eta_i}(F)$ *est un point fixe de* F*. Donc* $\sup_{i\in I} \mathrm{lfp}_{\eta_i}(F)$ *est aussi un point fixe de* F*. Il est supérieur ou égal à* $\sup_{i\in I} \eta_i$*, puisque* $\mathrm{lfp}_{\eta_i}(F) \geq \eta_i$ *pour tout* $i \in I$*. En tant que point fixe de* F *supérieur ou égal à* η*,* $\sup_{i\in I} \mathrm{lfp}_{\eta_i}(F)$ *est donc supérieur ou égal au plus petit,* $\mathrm{lfp}_\eta(F)$*.*

On peut en déduire que $Q[\![c]\!]$ est Scott-continue, par récurrence sur la taille de c, en faisant une récurrence auxiliaire pour démontrer que $Q[\![e]\!]$ est Scott-continue pour toute expression e. Le cas où e est une addition est traité par la question 6, le cas où c'est un opposé par la question 7. Le cas où c est une boucle while est traité par la question 8. Nous ne demandons pas de faire la démonstration complète, et admettrons dans la suite que $Q[\![c]\!]$ est Scott-continue.

Nous utilisons désormais une sémantique opérationnelle à petits pas de IMP—la première des notes de cours. Les règles sont en figure 1 ; ρ y dénote un environnement (réel).

Pour toute commande c, et tout ensemble $\mathcal{E}$ d'environnements (réels), disons que l'environnement ρ' est *accessible* depuis c et $\mathcal{E}$ si et seulement s'il existe un environnement (réel) ρ, dans $\mathcal{E}$, tel que $(c, \rho) \to^* \rho'$ ou bien $(c, \rho) \to^* (c', \rho')$ pour une certaine commande c'. On

notera $X[\![c]\!]\mathcal{E}$ l'ensemble des environnements accessibles depuis c et $\mathcal{E}$. On a les relations :

$$\begin{aligned}
X[\![x := e]\!]\mathcal{E} &= \mathcal{E} \cup \{\rho[x \mapsto [\![e]\!]\rho] \mid \rho \in \mathcal{E}\} && (6)\\
X[\![\texttt{skip}]\!]\mathcal{E} &= \mathcal{E} && (7)\\
X[\![c_1; c_2]\!]\mathcal{E} &= X[\![c_2]\!](X[\![c_1]\!]\mathcal{E}) && (8)\\
X[\![\texttt{if } e \texttt{ then } c_1 \texttt{ else } c_2]\!]\mathcal{E} &= \mathcal{E} \cup X[\![c_2]\!]\{\rho \in \mathcal{E} \mid [\![e]\!]\rho = 0\} && (9)\\
&\quad \cup X[\![c_1]\!]\{\rho \in \mathcal{E} \mid [\![e]\!]\rho \neq 0\}\\
X[\![\texttt{while } e \texttt{ do } c]\!]\mathcal{E} &= \mathrm{lfp}_{\mathcal{E}}(\Phi_{e,c}) && (10)
\end{aligned}$$

où $\Phi_{e,c}$ est la fonction définie par :

$$\Phi_{e,c}(\mathcal{E}') = \mathcal{E}' \cup X[\![c]\!]\{\rho \in \mathcal{E}' \mid [\![e]\!]\rho \neq 0\}$$

et l'on rappelle que $\mathrm{lfp}_{\mathcal{E}}$ désigne l'opérateur plus petit point fixe au-dessus de $\mathcal{E}$. Ici, $\mathcal{E}$, $\mathcal{E}'$ appartiennent au treillis complet dcpo $\mathbb{P}(Env)$ des ensembles d'environnements (réels), ordonné par inclusion $\subseteq$.

On ne demandera pas de démontrer ces égalités… sauf la dernière. C'est le sujet des deux questions qui viennent.

9. Montrer que la fonction $filt\colon \mathcal{E}' \mapsto \{\rho \in \mathcal{E}' \mid [\![e]\!]\rho \neq 0\}$ est Scott-continue. On pourra utiliser sans preuve que le supremum $\sup_{i\in I} \mathcal{E}_i$ dans $\mathbb{P}(Env)$ est l'union $\bigcup_{i\in I} \mathcal{E}_i$.

   Si $\mathcal{E}'_1 \subseteq \mathcal{E}'_2$, alors tout $\rho \in \mathcal{E}'_1$ tel que $[\![e]\!]\rho \neq 0$ est un $\rho \in \mathcal{E}'_2$ tel que $[\![e]\!]\rho \neq 0$, donc $filt$ est monotone.
   Ensuite, soit $(E_i)_{i\in I}$ une famille dirigée dans $\mathbb{P}(Env)$, de sup $\mathcal{E} = \bigcup_{i\in I} \mathcal{E}_i$. Alors $filt(\mathcal{E}) = \{\rho \mid \exists i \in I. \rho \in \mathcal{E}_i$ et $[\![e]\!]\rho \neq 0\} = \bigcup_{i\in I}\{\rho \mid \rho \in \mathcal{E}_i$ et $[\![e]\!]\rho \neq 0\} = \sup_{i\in I} filt(\mathcal{E}_i)$.

Au vu de la définition de $X[\![c]\!]\mathcal{E}$ comme ensemble de traces accessibles, il n'est pas trop difficile de démontrer que $X[\![c]\!]$ est Scott-continue. Comme toute composée de fonctions Scott-continues est Scott-continue, et que la fonction $\cup$ est trivialement Scott-continue, on en déduit que $\Phi_{e,c}$ est elle aussi Scott-continue. Je n'en demande pas de démonstration plus détaillée.

10. On admet qu'on peut démontrer que $X[\![\texttt{while } e \texttt{ do } c]\!]\mathcal{E}$ est égal à $\mathcal{E}_0 \cup \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_n \cup \cdots$, où la suite $\mathcal{E}_n$ est définie par :

$$\begin{aligned}
\mathcal{E}_0 &= \mathcal{E}\\
\mathcal{E}_{n+1} &= \mathcal{E}_n \cup X[\![c]\!]\{\rho \in \mathcal{E}_n \mid [\![e]\!]\rho \neq 0\}
\end{aligned}$$

   A partir de cette observation, démontrer l'égalité (10).

   C'est la formule de Kleene : $X[\![\texttt{while } e \texttt{ do } c]\!]\mathcal{E} = \bigcup_{n\in\mathbb{N}} \Phi^n_{e,c}(\emptyset)$. C'est aussi égal à $\bigcup_{n\geq 1} \Phi^n_{e,c}(\emptyset)$, puisque l'ensemble vide ne contribue pas à l'union. Par récurrence sur $n \in \mathbb{N}$, on montre que $\Phi^{n+1}_{e,c}(\emptyset) = \mathcal{E}_n$. D'où le résultat.

Nous pouvons désormais admettre la validité des équations (6) à (10).

On définit la fonction γ de $\mathcal{I}$ vers $\mathbb{P}(\mathbb{Z})$: $\gamma(\bot) = \emptyset$, $\gamma((a, b))$ est l'intervalle entier $]a, b[$ (ouvert, et non vide).

Réciproquement, pour toute partie E de $\mathbb{Z}$, on définit $\alpha(E) \in \mathcal{I}$ par : $\alpha(\emptyset) = \bot$, et si E est non vide, alors $\alpha(E) = (\inf E - 1, \sup E + 1)$. (Les infs et sups sont de nouveau calculés dans $\mathbb{Z} \cup \{-\infty, +\infty\}$, et $(-\infty) - 1 = (-\infty)$, $(+\infty) + 1 = +\infty$.)

On vérifie aisément que α et γ sont deux fonctions monotones, lorsque $\mathbb{P}(\mathbb{Z})$ est ordonné par l'ordre d'inclusion $\subseteq$. On peut aussi vérifier que $\alpha(\gamma(v)) = v$ pour tout $v \in \mathcal{I}$ et que $E \subseteq \gamma(\alpha(E))$ pour toute partie E de $\mathbb{Z}$.

11. En déduire que, pour tout $v \in \mathcal{I}$, pour tout $E \in \mathbb{P}(\mathbb{Z})$, $\alpha(E) \leq v$ ssi $E \subseteq \gamma(v)$.

    On peut le faire par énumération de cas, mais c'est vraiment pénible... et j'ai demandé de le « déduire » des affirmations précédentes, pas de le démontrer.

    Plus simplement : si $\alpha(E) \leq v$ alors $E \subseteq \gamma(\alpha(E))$ [dernière des affirmations précédentes] $\subseteq \gamma(v)$ [γ est monotone]. Réciproquement, si $E \subseteq \gamma(v)$ alors $\alpha(E) \leq \alpha(\gamma(v))$ [α est monotone] $= v$ [première des affirmations].

    Si l'une de ces inégalités équivalentes est vérifiée, on dira que v *représente correctement* l'ensemble E.

On généralise cela aux environnements : un environnement quotient η représente correctement un ensemble $\mathcal{E}$ d'environnements (réels) si et seulement si $\eta(x)$ représente correctement $\{\rho(x) \mid \rho \in \mathcal{E}\}$ pour toute variable x.

On admettra que si η représente correctement $\mathcal{E}$, alors pour toute expression e, $Q[\![e]\!]\eta$ représente correctement $\{[\![e]\!]\rho \mid \rho \in \mathcal{E}\}$.

12. Soit c une commande, et supposons que pour tout environnement quotient η' et tout ensemble $\mathcal{E}'$ d'environnements (réels) tels que η' représente correctement $\mathcal{E}'$, $Q[\![c]\!]\eta'$ représente correctement $X[\![c]\!]\mathcal{E}'$.

    Montrer que, pour tous η et $\mathcal{E}$, si η représente correctement $\mathcal{E}$, alors $Q[\![\texttt{while}\, e\, \texttt{do}\, c]\!]\eta$ représente correctement $X[\![\texttt{while}\, e\, \texttt{do}\, c]\!]\mathcal{E}$.

    1. On va commencer par démontrer que si η' représente correctement $\mathcal{E}'$, alors $F_{e,c}(\eta')$ représente correctement $\Phi_{e,c}(\mathcal{E}')$.

    Si $Q[\![e]\!]\eta' = \bot$ ou $Q[\![e]\!]\eta' = (-1, 1)$, par le fait admis énoncé juste avant la question, $Q[\![e]\!]\eta'$ représente correctement $E = \{[\![e]\!]\rho \mid \rho \in \mathcal{E}\}$. Donc E est inclus dans $\emptyset$ ou dans $\gamma((-1, 1)) = \{0\}$, selon le cas. Dans tous les cas, tous les éléments de E (s'il y en a) sont nuls. On a donc que $X[\![c]\!]\{\rho \in \mathcal{E}' \mid [\![e]\!]\rho \neq 0\} = \emptyset$, donc $\Phi_{e,c}(\mathcal{E}') = \mathcal{E}'$. Donc $F_{e,c}(\eta') = \eta'$ représente correctement $\Phi_{e,c}(\mathcal{E}') = \mathcal{E}'$.

    Si $Q[\![e]\!]\eta' \neq \bot$ et $Q[\![e]\!]\eta' \neq (-1, 1)$, alors $F_{e,c}(\eta') = \eta' \vee Q[\![c]\!]\eta'$. Par hypothèse, η' représente correctement $\mathcal{E}'$, c'est-à-dire que pour toute variable x, $\mathcal{E}'(x) \subseteq \gamma(\eta'(x))$. Par l'hypothèse faite en début de question, $Q[\![c]\!]\eta'$ représente correctement $X[\![c]\!]\mathcal{E}'$, donc pour toute variable x, $X[\![c]\!]\mathcal{E}'(x) \subseteq \gamma(Q[\![c]\!]\eta'(x))$. Donc $\Phi_{e,c}(\mathcal{E}')(x) = \mathcal{E}'(x) \cup X[\![c]\!]\mathcal{E}'(x) \subseteq \gamma(\eta'(x)) \cup \gamma(Q[\![c]\!]\eta'(x))$.

    Pour obtenir le résultat désiré $\Phi_{e,c}(\mathcal{E}')(x) \subseteq \gamma(\eta'(x) \vee Q[\![c]\!]\eta'(x)) = \gamma(F_{e,c}(\eta')(x))$, il ne reste qu'à démontrer l'inégalité $\gamma(v_1) \cup \gamma(v_2) \subseteq \gamma(v_1 \vee v_2)$. C'est une conséquence de la monotonie de γ, qui implique que $\gamma(v_1)$ et $\gamma(v_2)$ sont tous les deux inclus dans $\gamma(v_1 \vee v_2)$.

    (On peut aussi y arriver en passant par α, mais c'est plus compliqué : il faudra démontrer $\alpha(E_1 \cup E_2) \leq \alpha(E_1) \vee \alpha(E_2)$ pour $E_1 = \mathcal{E}'(x)$ et $E_2 = X[\![c]\!]\mathcal{E}'(x)$. Ceci ne découle pas de la monotonie de α, mais du fait que α préserve les sups [binaires,

ici]. En effet, $\alpha(E_1 \cup E_2) \leq \alpha(E_1) \vee \alpha(E_2)$ ssi $E_1 \cup E_2 \subseteq \gamma(\alpha(E_1) \vee \alpha(E_2))$ ssi E_1 et E_2 sont inclus dans $\gamma(\alpha(E_1) \vee \alpha(E_2))$, ce qui est vrai parce que $E_i \subseteq \gamma(\alpha(E_i)) \subseteq \gamma(\alpha(E_1) \vee \alpha(E_2))$.)

2. On veut démontrer que $\mathrm{lfp}_\eta(F_{e,c})$ représente correctement $\mathrm{lfp}_{\mathcal{E}}(\Phi_{e,c})$. Il y a plusieurs façons de le faire.

La plus évidente peut-être, mais l'une des plus longues, est d'utiliser la formule de Kleene : $\mathrm{lfp}(F) = \sup_{n\in\mathbb{N}} F^n(\bot)$ pour F Scott-continue sur un dcpo pointé, et de démontrer par récurrence sur n que $F^n_{e,c}(\eta')$ représente correctement $\Phi^n_{e,c}(\mathcal{E}')$ pour tout $n \in \mathbb{N}$. Ceci présente quelques difficultés : on doit appliquer la formule de Kleene à $F'_{e,c} \colon \eta' \mapsto \eta \vee F_{e,c}(\eta')$ et à $\Phi'_{e,c} \colon \mathcal{E}' \mapsto \mathcal{E} \cup \Phi_{e,c}(\mathcal{E}')$, et rappeler pourquoi ces deux fonctions sont Scott-continues.

Une autre façon de faire consiste à utiliser la question 10, et à démontrer que pour tout $n \in \mathbb{N}$, $\mathrm{lfp}_\eta(F_{e,c})$ représente correctement $\mathcal{E}_n$. Ceci revient à démontrer que $\mathcal{E}_n(x) \subseteq \gamma(\mathrm{lfp}_\eta(F_{e,c})(x))$ pour toute variable x, ce dont on tirera que $\bigcup_{n\in\mathbb{N}} \mathcal{E}_n(x) = X[\![\texttt{while}\, e\, \texttt{do}\, c]\!]\mathcal{E}(x)$ est inclus dans $\gamma(\mathrm{lfp}_\eta(F_{e,c})(x))$, autrement dit que $\mathrm{lfp}_\eta(F_{e,c}) = Q[\![\texttt{while}\, e\, \texttt{do}\, c]\!]\eta$ représente correctement $X[\![\texttt{while}\, e\, \texttt{do}\, c]\!]\mathcal{E}(x)$.

Ceci se fait par récurrence sur n. Pour $n = 0$, on doit démontrer que $\mathcal{E}(x) \subseteq \gamma(\mathrm{lfp}_\eta(F_{e,c})(x))$ pour toute variable x. Or $\mathcal{E}(x) \subseteq \gamma(\eta(x))$ puisque η représente correctement $\mathcal{E}$, et le fait que $\eta(x) \leq \mathrm{lfp}_\eta(F_{e,c})(x)$ plus la monotonie de γ suffisent à établir la conclusion souhaitée. A l'indice $n+1$, on doit démontrer que $\mathcal{E}_{n+1}(x) \subseteq \gamma(\mathrm{lfp}_\eta(F_{e,c})(x))$ sachant que $\mathrm{lfp}_\eta(F_{e,c})$ représente correctement $\mathcal{E}_n$. Or par le point 1 ci-dessus, $F_{e,c}(\mathrm{lfp}_\eta(F_{e,c}))$, qui est égal à $\mathrm{lfp}_\eta(F_{e,c})$ par définition d'un point fixe, représente aussi correctement $\Phi_{e,c}(\mathcal{E}_n) = \mathcal{E}_{n+1}$, ce qui termine la démonstration.

13. La similarité entre la définition de $Q[\![c]\!]\rho$ est les équations caractérisant $X[\![c]\!]\mathcal{E}$ nous permettent de démontrer que pour tous η et $\mathcal{E}$, si η représente correctement $\mathcal{E}$, alors $Q[\![c]\!]\eta$ représente correctement $X[\![c]\!]\mathcal{E}$. Nous avons déjà traité le cas de la boucle `while` à la question 12. Cette démonstration s'effectue par récurrence, mais sur quoi ?

    Sur la structure (ou la taille) de c.

14. Poursuivons la question précédente. Le lemme à prouver dans le cas de la boucle `while` a été décrit à la question 12. Dans le cas de l'affectation, quel lemme devons-nous prouver ?

    On doit montrer que si η représente correctement $\mathcal{E}$, alors $\eta \vee \eta[x \mapsto Q[\![\eta]\!]]$ $(= Q[\![x := e]\!]\eta)$ représente correctement $\mathcal{E} \cup \{\rho[x \mapsto [\![e]\!]\rho \mid \rho \in \mathcal{E}\}$ $(= X[\![x := e]\!]\mathcal{E})$.

15. Prouvez ce lemme.

    On rappelle que $Q[\![e]\!]\eta$ représente correctement $\{[\![e]\!]\rho \mid \rho \in \mathcal{E}\}$, c'est-à-dire : pour tout $\rho \in \mathcal{E}$, $[\![e]\!]\rho$ appartient à $\gamma(Q[\![e]\!]\eta)$. On regarde chaque variable, et deux cas se présentent, selon que c'est x ou une autre variable.

    Pour x, on doit démontrer que $\eta(x) \vee Q[\![\eta]\!](x)$ représente correctement $\{\rho(x) \mid \rho \in \mathcal{E}\} \cup \{[\![e]\!]\rho \mid \rho \in \mathcal{E}\}$, autrement dit que pour tout $\rho \in \mathcal{E}$, $\rho(x)$ et $[\![e]\!]\rho$ sont tous les deux dans $\gamma(\eta(x) \vee Q[\![\eta]\!](x))$. Or comme η représente correctement $\mathcal{E}$, et que $\rho \in \mathcal{E}$, $\rho(x)$ est dans $\gamma(\eta(x))$; et $[\![e]\!]\rho$ est dans $\gamma(Q[\![\eta]\!](x))$. Tant $\gamma(\eta(x))$ que $\gamma(Q[\![\eta]\!](x))$ est inclus dans $\gamma(\eta(x) \vee Q[\![\eta]\!](x))$, ce qui nous permet de conclure. (Un autre argument consisterait à dire que η représente correctement $\mathcal{E}$ et $\eta[x \mapsto Q[\![e]\!]\eta]$ représente correctement $\{\rho[x \mapsto [\![\rho]\!] \mid \rho \in \mathcal{E}\}$, donc le sup des premiers représente

correctement l'union des seconds. Ceci se démontre à l'aide de la monotonie de γ, ou bien du fait que α préserve les sups binaires, ce qu'il faut alors démontrer.)

Pour toute variable $y \neq x$, on doit démontrer que $\eta(y)$ représente correctement $\{\rho(y) \mid \rho \in \mathcal{E}\} \cup \{\rho[x \mapsto [\![e]\!]\rho](y) \mid \rho \in \mathcal{E}\}$, c'est-à-dire $\{\rho(y) \mid \rho \in \mathcal{E}\}$. Ceci découle directement du fait que η représente correctement $\mathcal{E}$.

Les cas des autres constructions du langages ne seront pas traitées ici. Tout fonctionne : si η représente correctement $\mathcal{E}$, alors $Q[\![c]\!]\eta$ représente correctement $X[\![c]\!]\mathcal{E}$.

16. On peut penser calculer $Q[\![c]\!]\eta$ par récurrence sur la structure de c, en suivant les équations (1) à (5). Pour cette dernière, on peut calculer $\mathrm{lfp}_\eta(F_{e,c})$ par *itérations de Kleene* : comme la borne supérieure de η, $F_{e,c}(\eta)$, $F^2_{e,c}(\eta)$, etc., en calculant $\eta_0 = \eta$, $\eta_{n+1} = F_{e,c}(\eta_n)$, et en s'arrêtant à la première étape n où $\eta_{n+1} = \eta_n$. Montrer que ceci ne termine pas, en exhibant un programme `while` e `do` c adéquat.

Par exemple `while` $\dot{1}$ `do` $x := x \dot{+} \dot{1}$ *en partant de $\eta_0 : x \mapsto (-1, 1)$ fabrique des η_n qui envoient x vers $(-1, n+1)$ (si j'avais décrit l'addition dans $\mathcal{I}$ correctement ; $(-1, 2n+1)$ avec la définition donnée). Le sup est $(-1, +\infty)$, qui n'est pas atteint.*

17. Pour corriger ce problème, je propose de remplacer $F_{e,c}$ dans (5) par $F'_{e,c}$ (autrement dit je remplace la définition de $Q[\![\texttt{while}\, e\, \texttt{do}\, c]\!]\eta$ par $\mathrm{lfp}_\eta(F'_{e,c})$), définie comme suit :

$$F'_{e,c}(\eta') \quad = \quad \begin{cases} \eta' & \text{si } Q[\![e]\!]\eta' = \bot \text{ ou } Q[\![e]\!]\eta' = 0 \\ \eta' \nabla Q[\![c]\!]\eta' & \text{sinon.} \end{cases},$$

où l'opération ∇ est définie par : $\bot \nabla v = v$, $v \nabla \bot = v$, $(a, b)\nabla(c, d)$ est égal à (a', b') où $a' = a$ si $c \geq a$, $a' = -\infty$ sinon, et $b' = b$ si $d \leq b$, $b' = +\infty$ sinon.

Pourquoi le calcul de $\mathrm{lfp}_\eta(F'_{e,c})$ par itérations de Kleene termine-t-il ? Pourquoi, avec la nouvelle définition, $Q[\![c]\!]\eta$ représente-t-il toujours correctement $X[\![c]\!]\mathcal{E}$ pour toute commande c ?

On étend bien sûr ∇ des valeurs aux environnements, composante par composante.

1. L'idée générale est qu'à chaque itération de point fixe qui n'a pas stabilisé, on a rendu au moins une des composantes du couple de plus égale à un infini.

Formellement, définissons $i(\bot) = 0$, $i((a, b)) = 1$ si $a \neq -\infty$ et $b \neq +\infty$, $i((-\infty, b)) = 2$ si $b \neq +\infty$, $i((a, +\infty)) = 2$ si $a \neq -\infty$, et $i((-\infty, +\infty)) = 3$. On montre que $v_1 \nabla v_2$ est toujours supérieur ou égal à v_1, et s'il lui est supérieur alors $i(v_1) < i(v_1 \nabla v_2)$. Le seul cas important à vérifier est celui où $v_1 = (a, b)$, $v_2 = (c, d)$, et $v_1 \neq v_1 \nabla v_2$. (Posons $v_1 \nabla v_2 = (a', b')$.) On ne peut donc pas avoir $c \geq a$ et $d \leq b$. Si $c < a$, alors a était différent de $-\infty$ et $a' = -\infty$, créant un infini de plus. Si $d > b$, alors b était différent de $+\infty$ et $b' = +\infty$, créant aussi un infini de plus.

Comme les valeurs de i varient entre 0 et 3, une suite $v_{n+1} = v_n \nabla v'_n$ ne peut croître strictement qu'au plus 3 fois.

Pour des environnements sur N variables, une suite $\eta_{n+1} = \eta_n \nabla \eta'_n$ ne peut croître strictement qu'au plus $3N$ fois. Or la suite des itérés de Kleene de $F'_{e,c}$ est justement de cette forme, et stabilise donc.

2. Pour montrer que $Q[\![c]\!]\eta$ représente toujours correctement $X[\![c]\!]\mathcal{E}$ avec la nouvelle définition (lorsque η représente correctement $\mathcal{E}$), il suffit d'observer que : $v \nabla v' \geq v \vee v'$, donc $\eta' \nabla Q[\![c]\!]\eta' \geq \eta' \vee Q[\![c]\!]\eta'$.

Ceci permet de montrer par récurrence sur c que $Q[\![c]\!]\eta$ avec la nouvelle définition est toujours supérieure ou égal à $Q[\![c]\!]\eta$ avec l'ancienne définition. On utilise pour cela par ailleurs que l'ancienne définition de $Q[\![c]\!]\eta$ est monotone en η, par exemple dans le cas où c est une affectation : $(*)$ $Q_{nouvelle}[\![c_1; c_2]\!]\eta = Q_{nouvelle}[\![c_2]\!](Q_{nouvelle}[\![c_1]\!]\eta)$ est supérieur ou égal à $Q_{ancienne}[\![c_2]\!](Q_{nouvelle}[\![c_1]\!]\eta)$, qui est donc supérieur ou égal à $Q_{ancienne}[\![c_2]\!](Q_{ancienne}[\![c_1]\!]\eta)$.

Comme l'ancienne définition représente correctement $X[\![c]\!]\eta$, la nouvelle aussi. Ceci est parce que si v représente correctement E ($\alpha(E) \leq v$) et $v' \geq v$, alors $\alpha(E) \leq v'$.

Comme certains l'ont remarqué, ∇ n'est pas monotone (elle l'est en son second argument, mais pas en le premier), ce qui implique que $Q[\![c]\!]\eta$, dans la nouvelle définition, n'est pas monotone en η. Ceci implique un certain nombre de choses, dont la seule un peu gênante est que $F'_{e,c}$ n'a pas nécessairement de point fixe, et que j'aurais dû réécrire explicitement la nouvelle sémantique quotient des boucles while comme un sup d'itérés de $F'_{e,c}$.

D'autre part, l'argument $(*)$ ne peut pas être remplacé par l'argument (faux) suivant : « $Q_{nouvelle}[\![c_1; c_2]\!]\eta = Q_{nouvelle}[\![c_2]\!](Q_{nouvelle}[\![c_1]\!]\eta)$ est supérieur ou égal à $Q_{nouvelle}[\![c_2]\!](Q_{ancienne}[\![c_1]\!]\eta)$ (faux : la nouvelle sémantique n'est plus monotone), qui est donc supérieur ou égal à $Q_{ancienne}[\![c_2]\!](Q_{ancienne}[\![c_1]\!]\eta)$ ».

C'est un exemple simple d'*interprétation abstraite*, une technique d'analyse statique de programmes, donnant des informations sur les valeurs calculées par le programme sans avoir à l'exécuter.