Corrigé de la composition d'informatique-mathématiques ENS - MP - 2020

Stéphane Legros, Yann Salmon

29 décembre 2023

Remarque : pour alléger les notations, nous nous permettrons quelquefois de noter $\leq$ (resp. $\sup_{i\in\mathbb{N}}$) au lieu de $\leq_\tau$ (resp. $\sup^A_{i\in\mathbb{N}}$). D'autre part, comme dans l'exemple du bas de la page 4, il faut comprendre que le type $\tau_1 \Rightarrow \tau_2 \Rightarrow \tau_3$ est en fait $\tau_1 \Rightarrow (\tau_2 \Rightarrow \tau_3)$ (convention usuelle en Ocaml).

1 Partie I

1. (a) Les suites croissantes de $(\llbracket\mathsf{nat}\rrbracket, \leq_{\mathsf{nat}})$ sont la suite constante $(\bot_{\mathsf{nat}})_{i\in\mathbb{N}}$ et les suites $(a_i)_{i\in\mathbb{N}}$ telles qu'il existe $i_0 \in \mathbb{N}$ et $k \in \mathbb{N}$ avec $a_i = \bot_{\mathsf{nat}}$ pour $i < i_0$ et $a_i = k$ pour $i \geq i_0$.

   (b) La suite constante $(\bot_{\mathsf{nat}})_{i\in\mathbb{N}}$ admet $\bot_{\mathsf{nat}}$ pour borne supérieure et une suite croissante de la seconde forme admet k pour borne supérieure : $(\llbracket\mathsf{nat}\rrbracket, \leq_{\mathsf{nat}})$ est un ensemble ordonné complet.

2. (a) Soit $f : \llbracket\mathsf{nat}\rrbracket \to \llbracket\mathsf{unit}\rrbracket$ une application croissante et $(a_i)_{i\in\mathbb{N}}$ une suite croissante de $(\llbracket\mathsf{nat}\rrbracket, \leq_{\mathsf{nat}})$. Les suites $(a_i)_{i\in\mathbb{N}}$ et $(f(a_i))_{i\in\mathbb{N}}$ sont croissantes et stationnaires (resp. en k et $f(k)$ avec $k \in \llbracket\mathsf{nat}\rrbracket$) : leurs bornes supérieures sont les valeurs auxquelles elles stationnent, donc $f(\sup_{i\in\mathbb{N}} a_i) = f(k) = \sup_{i\in\mathbb{N}} f(a_i)$ et f est continue.

   (b) Soit $f \in \llbracket\mathsf{nat} \Rightarrow \mathsf{unit}\rrbracket$ tel que $f(\bot_{\mathsf{nat}}) \neq \bot_{\mathsf{unit}}$. On a donc $f(\bot_{\mathsf{nat}}) = \mathsf{uu}$, puis

   $$\forall k \in \mathbb{N}, \mathsf{uu} = f(\bot_{\mathsf{nat}}) \leq f(k)$$

   par croissance de f : ceci impose $f(k) = \mathsf{uu}$ et f est constante. Il y a donc une unique telle fonction, car la fonction constante convient bien.

   (c) Pour $i \in \mathbb{N}$, on pose $f_i : k \longmapsto \begin{cases} \mathsf{uu} & \text{si } k \in \{0, 1, \ldots, i\} \\ \bot_{\mathsf{unit}} & \text{sinon} \end{cases}$.

3. (a) Soient $a, b \in \llbracket\tau\rrbracket$ avec $a \leq b$. On a $f_i(a) \leq f_i(b)$ pour tout $i \in \mathbb{N}$, donc $f'(a) = \sup_{i\in\mathbb{N}} f_i(a) \leq \sup_{i\in\mathbb{N}} f_i(b) = f'(b)$. Ainsi f' est croissante.
   Soit $(e_n)_{n\in\mathbb{N}}$ une suite croissante de $\llbracket\tau_1\rrbracket$ et $e = \sup_{n\in\mathbb{N}} e_n$. Nous avons $e_n \leq e$ pour tout $n \in \mathbb{N}$, donc, par croissance de f', $f'(e_n) \leq f'(e)$ pour tout $n \in \mathbb{N}$, puis $\sup_{n\in\mathbb{N}} f'(e_n) \leq f'(e)$.
   Soit $\alpha \in \llbracket\tau_2\rrbracket$ un majorant de $(f'(e_n))_{n\in\mathbb{N}}$, on a :

   $$\forall i \in \mathbb{N}, \forall n \in \mathbb{N}, f_i(e_n) \leq \alpha.$$

   On en déduit, par continuité des f_i :

   $$\forall i \in \mathbb{N}, f_i(e) = f_i(\sup_{n\in\mathbb{N}} e_n) = \sup_{n\in\mathbb{N}} f_i(e_n) \leq \alpha$$

   et donc $f'(e) \leq \alpha$. Ainsi $f'(e)$ est le plus petit des majorants de la suite $(f'(e_n))_{n\in\mathbb{N}}$ et f' est continue.

(b) Soit $(f_i)_{i\in\mathbb{N}}$ une suite croissante de $[\![\tau_1 \Rightarrow \tau_2]\!]$ (pour l'ordre $\leq_{\tau_1\Rightarrow\tau_2}$) et soit f' la fonction définie à la question précédente (qui établit que $f' \in [\![\tau_1 \Rightarrow \tau_2]\!]$). On a :

$$\forall i \in \mathbb{N}, \forall e \in [\![\tau_1]\!], f_i(e) \leq_{\tau_2} f'(e)$$

donc f' est un majorant de $(f_i)_{i\in\mathbb{N}}$ pour l'ordre $\leq_{\tau_1\Rightarrow\tau_2}$.
Si $g \in [\![\tau_1 \Rightarrow \tau_2]\!]$ majore la suite $(f_i)_{i\in\mathbb{N}}$, nous avons :

$$\forall i \in \mathbb{N}, \forall e \in [\![\tau_1]\!], f_i(e) \leq_{\tau_2} g(e)$$

et donc

$$\forall e \in [\![\tau_1]\!], f'(e) = \sup_{i\in\mathbb{N}} f_i(e) \leq_{\tau_2} g(e)$$

donc $f' \leq_{\tau_1\Rightarrow\tau_2} g$: f' est la borne supérieure de $(f_i)_{i\in\mathbb{N}}$ et $([\![\tau_1 \Rightarrow \tau_2]\!], \leq_{\tau_1\Rightarrow\tau_2})$ est un ordre partiel complet.

2 Partie II

1. Le programme proposé est bien typé à la condition que $t(x) = t(y) = t(z)$ et que ce type soit dans $\{\mathsf{unit}, \mathsf{bool}, \mathsf{nat}\}$; il est en ce cas de type $t(x) \Rightarrow t(x)$.
2. (a) Il suffit d'utiliser les définitions ; on a successivement, pour tout $n \in \mathbb{N}$:
   — $[\![x]\!]^{\mathcal{E}[x\mapsto n]} = n$;
   — $[\![x/2]\!]^{\mathcal{E}[x\mapsto n]} = \lfloor n/2 \rfloor$ et $[\![2]\!]^{\mathcal{E}[x\mapsto n]} = 2$;
   — $[\![2 \times (x/2)]\!]^{\mathcal{E}[x\mapsto n]} = 2\lfloor n/2 \rfloor$;
   — $[\![(2 \times (x/2)) = x]\!]^{\mathcal{E}[x\mapsto n]} = \begin{cases} \mathsf{tt} & \text{si } 2\lfloor n/2 \rfloor = n \\ \mathsf{ff} & \text{si } 2\lfloor n/2 \rfloor \neq n \end{cases}$

   et donc

   $$[\![p_e]\!]^{\mathcal{E}}(n) = [\![(2 \times (x/2)) = x]\!]^{\mathcal{E}[x\mapsto n]} = \begin{cases} \mathsf{tt} & \text{si } n \text{ est pair} \\ \mathsf{ff} & \text{si } n \text{ est impair} \end{cases}.$$

   L'ensemble cherché est donc l'ensemble des entiers naturels pairs.
   (b) On propose $p_c = \mathsf{fun}\, x \to (\mathsf{if}\ (p_e\ x)\ \mathsf{then}\ x/2\ \mathsf{else}\ ((3 \times x) + 1))$.
3. (a) On a

   $$[\![p]\!](f)(e) = \begin{cases} \bot_{\mathsf{nat}} & \text{si } e = \bot_{\mathsf{nat}} \\ 1 & \text{si } e = 0 \\ \bot_{\mathsf{nat}} & \text{si } e \in \mathbb{N}^* \text{ et } f(e-1) = \bot_{\mathsf{nat}} \\ e \cdot f(e-1) & \text{sinon} \end{cases}.$$

   (b) Montrons par récurrence que pour tout $i \in \mathbb{N}$,

   $$\forall e \in [\![\mathsf{nat}]\!], [\![p]\!]^{i+1}(\bot_{\mathsf{nat}\Rightarrow\mathsf{nat}})(e) = \begin{cases} \bot_{\mathsf{nat}} & \text{si } e = \bot_{\mathsf{nat}} \\ e! & \text{si } e \in \{0, 1, \ldots, i\} \\ \bot_{\mathsf{nat}} & \text{sinon} \end{cases}.$$

   Pour $i = 0$, la propriété est vérifiée car $\bot_{\mathsf{nat}\Rightarrow\mathsf{nat}}(e-1) = \bot_{\mathsf{nat}}$ pour $e \geq 1$. Soit $i \in \mathbb{N}$ et supposons que la relation soit vraie au rang i. Soit $e \in [\![\mathsf{nat}]\!]$: $[\![p]\!]^{i+2}(\bot)(e) = [\![p]\!]\left([\![p]\!]^{i+1}(\bot)\right)(e)$, donc
   — si $e = \bot_{\mathsf{nat}}$, alors $[\![p]\!]^{i+2}(\bot)(e) = \bot_{\mathsf{nat}}$;
   — si $e = 0$, $[\![p]\!]^{i+2}(\bot)(e) = 1 = e!$;

— si $e \geq i+2$, alors $e-1 \geq i+1$, donc $[\![p]\!]^{i+1}(\bot)(e-1) = \bot_{\mathsf{nat}}$ (HR) puis $[\![p]\!]^{i+2}(\bot)(e) = \bot_{\mathsf{nat}}$;

— sinon, $e \in \{1, \ldots, i+1\}$, donc $[\![p]\!]^{i+1}(\bot)(e-1) = (e-1)! \neq \bot_{\mathsf{nat}}$ (HR) et donc $[\![p]\!]^{i+2}(\bot)(e) = e \cdot [\![p]\!]^{i+1}(\bot)(e-1) = e!$.

Récurrence établie.

(c) Pour $k \in \mathbb{N}$, la suite $\left([\![p]\!]^i(\bot)(k)\right)_{i\in\mathbb{N}}$ stationne à la valeur $k!$, donc $[\![\mathsf{rec}\, p]\!](k) = k!$ (comme démontré à la question **??**, la limite d'une suite croissante (f_i) d'éléments de $[\![\tau_1 \Rightarrow \tau_2]\!]$ est la fonction f' définie par $f'(e) = \sup_{i\in\mathbb{N}} f_i(e)$).

Comme $[\![p]\!]^{i+1}(\bot)(\bot_{\mathsf{nat}}) = \bot_{\mathsf{nat}}$ pour tout i, $[\![\mathsf{rec}\, p]\!](\bot_{\mathsf{nat}}) = \bot_{\mathsf{nat}}$.

4. (a) Le programme $q := \mathsf{fun}\, f \to \mathsf{fun}\, n \to \mathsf{if}\, n = 1\, \mathsf{then}\, \mathsf{uu}\, \mathsf{else}\, f(p_c n)$ convient. On a en effet, pour $g \in [\![\mathsf{nat} \Rightarrow \mathsf{unit}]\!]$ et $e \in [\![\mathsf{nat}]\!]$:

$$[\![q]\!](g)(e) \begin{cases} \mathsf{uu} & \text{si } e = 1 \\ \bot_{\mathsf{unit}} & \text{si } e = \bot_{\mathsf{nat}} \\ g(f_c(e)) & \text{sinon} \end{cases}$$

et la propriété demandée est alors évidente par récurrence.

(b) On pose $p := \mathsf{rec}\, q$. Pour $e \in \mathbb{N}^*$, $[\![q]\!]^i(e)$ vaut $\bot_{\mathsf{unit}}$ si $f_c^k(e) \neq 1$ pour tout $k \leq i$ et uu sinon. La suite $([\![q]\!]^i(e))_{i\in\mathbb{N}}$ va donc stationner à $\bot_{\mathsf{unit}}$ si $f_c^k(e) \neq 1$ pour tout $k \in \mathbb{N}$ et stationner à uu sinon. On en déduit que $[\![\mathsf{rec}\, q]\!](e) = \mathsf{uu}$ ssi il existe $k \in \mathbb{N}$, $f_c^k(e) = 1$.

5. (a) $[\![\mathsf{unit} \Rightarrow \mathsf{unit}]\!]$ contient les trois éléments :

$$g_1 : \alpha \longmapsto \bot_{\mathsf{unit}}, \qquad g_2 : \alpha \longmapsto \alpha, \qquad g_3 : \alpha \longmapsto \mathsf{uu},$$

et en fixant une variable x de type unit, ces trois fonctions sont calculées par les programmes

$$p_1 = \mathsf{fun}\, x \to \mathsf{DIV}_{\mathsf{unit}}, \qquad p_2 = \mathsf{fun}\, x \to x, \qquad p_3 = \mathsf{fun}\, x \to \mathsf{uu}.$$

(b) Comme l'ensemble des programmes est dénombrable, l'ensemble $\mathcal{P}$ des programmes p : $[\![\mathsf{nat} \Rightarrow \mathsf{unit}]\!]$ dont l'interprétation est indépendante de l'environnement est également dénombrable ; on en déduit que les éléments calculables de $[\![\mathsf{nat} \Rightarrow \mathsf{unit}]\!]$ sont en quantité au plus dénombrable (image de $\mathcal{P}$ par l'application $p \longmapsto [\![p]\!]$). Comme l'ensemble des applications de nat dans unit est plus que dénombrable (cet ensemble est équipotent à $\mathcal{P}(\mathsf{nat})$: il a la puissance du continu), il existe des éléments non calculables dans $[\![\mathsf{nat} \Rightarrow \mathsf{unit}]\!]$.

6. Le programme $\mathsf{fun}\, x \to \mathsf{fun}\, y \to \mathsf{if}\, x = \mathsf{uu}\, \mathsf{then}\, y\, \mathsf{else}\, \mathsf{DIV}_{\mathsf{unit}}$ convient.

7. (a) Soit $f \in [\![\mathsf{bool} \Rightarrow \mathsf{unit}]\!]$. On a $[\![p]\!](f) = [\![(x\ \mathsf{tt})\ ||_{\mathsf{u}}\ (x\ \mathsf{ff})]\!]^{[x\mapsto f]}$, ce qui vaut uu ssi l'un des deux termes s'évalue en uu. L'ensemble cherché est donc celui des fonctions qui prennent la valeur uu (en tt ou en ff).

(b) C'est l'ensemble des $f \in [\![\mathsf{nat} \Rightarrow \mathsf{unit}]\!]$ telles qu'il existe $n \in \mathbb{N}$ tel que $f(n) = \mathsf{uu}$.

3 Partie III

1. Si x est une variable de type τ, $[\![\tau]\!]$ et son complémentaire sont calculés respectivement par $\mathsf{fun}\, x \to \mathsf{uu}$ et

$\mathsf{fun}\, x \to \mathsf{DIV}_{\mathsf{unit}}$: $[\![\tau]\!]$ est donc un ouvert calculatoire et un fermé calculatoire.

2. D'après la question **??**, les ouverts calculatoires de $[\![\mathsf{unit}]\!]$ sont les parties $\emptyset$, $\{\mathsf{uu}\}$ et $[\![\mathsf{unit}]\!]$. La quatrième partie $\{\bot_{\mathsf{unit}}\}$ n'est pas un ouvert calculatoire car son indicatrice, faute d'être croissante, n'est pas dans $[\![\mathsf{unit} \Rightarrow \mathsf{unit}]\!]$.

3. Il existe deux programmes $p : \tau \Rightarrow \tau'$ et $q : \tau' \Rightarrow \mathsf{unit}$ tels que $f = [\![p]\!]$ et $\mathbf{1}_U = [\![q]\!]$, en notant $\mathbf{1}_U$ la fonction indicatrice d'une partie U. En fixant une variable x de type τ', on peut définir le programme $r := \mathsf{fun}\, x \to q\ (p\ x)$, qui est de type $\tau \Rightarrow \mathsf{unit}$ et on a, pour tout $e \in [\![\tau]\!]$:

$$[\![r]\!](e) = \mathsf{uu} \text{ ssi } [\![q]\!]([\![p]\!](e)) = \mathsf{uu} \text{ ssi } [\![p]\!](e) \in U \text{ ssi } f(e) \in U \text{ ssi } e \in f^{-1}(U)$$

donc $f^{-1}(U)$ est un ouvert calculatoire.

4. Soient U et V deux ouverts calculatoires de $[\![\tau]\!]$, $p, q : \tau \Rightarrow \mathsf{unit}$ deux programmes tels que $[\![p]\!] = \mathbf{1}_U$ et $[\![q]\!] = \mathbf{1}_V$ et x une variable de type τ. Les programmes $\mathsf{fun}\, x \to (p\ x\ ||_{\mathsf{u}}\ q\ x)$ et $\mathsf{fun}\, x \to (p\ x\ \&_{\mathsf{u}}\ q\ x)$ sont alors de type $\tau \Rightarrow \mathsf{unit}$ et calculent respectivement $U \cup V$ et $U \cap V$.

5. On pose $q := \mathsf{fun}\, f \to \mathsf{fun}\, k \to \mathsf{fun}\, e \to (p\ k\ e\ ||_{\mathsf{u}}\ f\ (k+1)\ e)$, de type $\tau' \Rightarrow \tau'$ avec $\tau' = \mathsf{nat} \Rightarrow \tau \Rightarrow \mathsf{unit}$. On a, pour $k \in \mathbb{N}$ et $e \in [\![\tau]\!]$:

$$\begin{cases} [\![q]\!](\bot)(k)(e) = \mathsf{uu} \text{ ssi } [\![p]\!](k)(e) = \mathsf{uu} \text{ ssi } e \in U_k \\ [\![q]\!]^{i+1}(\bot)(k)(e) = \mathsf{uu} \text{ ssi } [\![p]\!](k)(e) = \mathsf{uu} \text{ ou } [\![q]\!]^i(\bot)(k+1)(e) = \mathsf{uu} \end{cases}$$

et donc par récurrence :

$$\forall i \in \mathbb{N}, [\![q]\!]^i(\bot)(k)(e) = \mathsf{uu} \text{ ssi } \exists j \in \{k, k+1, \ldots, k+i-1\}, e \in U_j$$

puis

$$[\![\mathsf{rec}\, q]\!](k)(e) = \mathsf{uu} \text{ ssi } e \in \bigcup_{j \geq k} U_j.$$

Le programme $(\mathsf{rec}\, q)\ 0$ calcule donc $\bigcup_{j \geq 0} U_j$, qui est ainsi un ouvert calculatoire.

6. (a) On souhaite ici que $[\![\forall_\emptyset\ p]\!] = \mathsf{uu}$ pour tout programme $p : \tau \Rightarrow \mathsf{unit}$, puisque la propriété $\big(\forall e \in \emptyset, [\![p]\!](e) = \mathsf{uu}\big)$ est vraie pour tout tel programme p. On fixe donc une variable x de type $\tau \Rightarrow \mathsf{unit}$ et on définit $\forall_\emptyset := \mathsf{fun}\, x \to \mathsf{uu}$.

   (b) Si $p : \tau \Rightarrow \mathsf{unit}$, alors $[\![p]\!]$ est croissante. Or E contient $\bot_\tau$, donc la propriété $\forall e \in E, [\![p]\!](e) = \mathsf{uu}$ se ramène à $[\![p]\!](\bot_\tau) = \mathsf{uu}$. Le sens direct est évident ; pour le sens réciproque, on a pour $e \in E$, $\bot_\tau \leq_\tau e$, donc $\mathsf{uu} = [\![p]\!](\bot_\tau) \leq_{\mathsf{unit}} [\![p]\!](e)$, c'est-à-dire $[\![p]\!](e) = \mathsf{uu}$.

   En fixant une variable x de type $\tau \Rightarrow \mathsf{unit}$, le programme $\forall_E := \mathsf{fun}\, x \to x\ \mathsf{DIV}_\tau$ donne pour $p : \tau \Rightarrow \mathsf{unit}$

$$[\![\forall_E\ p]\!] = \mathsf{uu} \text{ ssi } [\![p\ \mathsf{DIV}_\tau]\!] = \mathsf{uu} \text{ ssi } [\![p]\!](\bot_\tau) = \mathsf{uu} \text{ ssi } \forall e \in E, [\![p]\!](e) = \mathsf{uu}.$$

   Toute partie contenant $\bot_\tau$ est donc calculatoirement compacte.

   (c) Soit A un fermé calculatoire de $[\![\tau]\!]$ et $p : \tau \Rightarrow \mathsf{unit}$ un programme tel que $\mathbf{1}_{[\![\tau]\!] \setminus A} = [\![p]\!]$. Si A contient $\bot_\tau$, il est calculatoirement compact d'après le (b). Sinon, $\bot_\tau \in [\![\tau]\!] \setminus A$ et $[\![p]\!](\bot_\tau) = \mathsf{uu}$. Par croissance de $[\![p]\!]$, on a $[\![p]\!](e) = \mathsf{uu}$ pour tout $e \in [\![\tau]\!]$. Par conséquent, $A = \emptyset$, qui est calculatoirement compact d'après le (a).

7. On montre par récurrence sur son cardinal qu'une partie finie de $\mathbb{N}$ est calculatoirement compacte :
   - $\emptyset$ est une partie calculatoirement compacte dans $[\![\mathsf{nat}]\!]$ d'après la question précédente.
   - soit $k \in \mathbb{N}$ et supposons que toute partie de cardinal k de $\mathbb{N}$ est calculatoirement compacte dans $[\![\mathsf{nat}]\!]$; si A est une partie de $\mathbb{N}$ de cardinal $k+1$, on fixe $n \in \mathbb{N}$ et B de cardinal k tels que $A = B \uplus \{n\}$. Comme B est calculatoirement compact, on peut définir un programme $\forall_B$. On pose ensuite $\forall_A = \mathsf{fun}\, f \to (f\, n) \&_{\mathsf{u}} \forall_B\, f$ et on a pour tout programme $p : \mathsf{nat} \Rightarrow \mathsf{unit}$:

$$[\![\forall_A\ p]\!] = \mathsf{uu} \text{ ssi } [\![p]\!](n) = \mathsf{uu} \text{ et } \forall e \in B, [\![p]\!](e) = \mathsf{uu} \text{ ssi } \forall e \in A, [\![p]\!](e) = \mathsf{uu}.$$

On montre la réciproque par l'absurde. Soit U une partie infinie de $\mathbb{N}$ qu'on suppose calculatoirement compacte : il y a un programme $\forall_U$.

Soit $\mathbf{i}_{\{0\}} := \mathsf{fun}\, e \to \mathsf{if}\, e = 0\ \mathsf{then}\ \mathsf{uu}\ \mathsf{else}\ \mathsf{DIV}_{\mathsf{unit}}$ un programme indicateur de $\{0\}$. Soit $p := \mathsf{fun}\, f \to \mathsf{fun}\, e \to \mathbf{i}_{\{0\}}\ \ e\ ||_{\mathsf{u}}\ f\ \ (e-1)$ de type $(\mathsf{nat} \Rightarrow \mathsf{unit}) \Rightarrow (\mathsf{nat} \Rightarrow \mathsf{unit})$. Soit d'une part $q_\infty := \mathsf{rec}\, p : \mathsf{nat} \Rightarrow \mathsf{unit}$ et d'autre part, pour chaque $k \in \mathbb{N}$, $q_k := p\ \ (p\ \ \ldots (\bot_{\mathsf{nat}\Rightarrow\mathsf{unit}}) \ldots)$ avec k applications de p. Par construction, la suite des $[\![q_k]\!]$ est croissante et par définition, $[\![q_\infty]\!] = \sup_{k\in\mathbb{N}} [\![q_k]\!]$. On montre par récurrence que pour chaque $k \in \mathbb{N}$, q_k est un programme indicateur de $\{0, \ldots, k-1\}$; on en déduit aussi que q_∞ est un programme indicateur de $\mathbb{N}$.

Soit $k \in \mathbb{N}$. Comme U est une partie infinie de $\mathbb{N}$, U n'est pas majorée, donc il existe $u \in U$ tel que $u \geq k$. Ainsi, $[\![q_k]\!](u) = \bot_{\mathsf{unit}}$. Par définition, on a donc $[\![\forall_U\ \ q_k]\!] = \bot_{\mathsf{unit}}$, c'est-à-dire $[\![\forall_U]\!]([\![q_k]\!]) = \bot_{\mathsf{unit}}$. Par continuité, on a donc $[\![\forall_U]\!]([\![q_\infty]\!]) = \bot_{\mathsf{unit}}$.

Mais d'autre part, pour tout $u \in U$, $[\![q_\infty]\!](u) = \mathsf{uu}$ car $U \subset \mathbb{N}$, donc $[\![\forall_U\ \ q_\infty]\!] = \mathsf{uu}$, c'est-à-dire $[\![\forall_U]\!]([\![q_\infty]\!]) = \mathsf{uu}$.

Contradiction.

4 Partie IV

1. La suite $(f_{s<k})_{k\in\mathbb{N}}$ est croissante et $f_s = \sup_{k\in\mathbb{N}} f_{s<k}$, donc par continuité, $\mathsf{uu} = [\![p]\!](f_s) = \sup_{k\in\mathbb{N}} [\![p]\!](f_{s<k})$, donc la suite $([\![p]\!](f_{s<k}))_{k\in\mathbb{N}}$ ne peut stationner à $\bot_{\mathsf{unit}}$. D'où le résultat.
2. *Il est tentant d'écrire que par définition,* $[\![p/b]\!](f_{s<k}) = [\![p]\!]([\![b :: f_{s<k}]\!])$*, mais ça n'a pas de sens parce que* $f_{s<k}$ *n'est pas un programme.* Remarquons que pour tout $s \in \mathbb{B}^{\mathbb{N}}$ et tout $k \in \mathbb{N}$, la fonction $f_{s<k}$ est calculée par le programme

$$q_{s<k} := \mathsf{fun}\, i \to \mathsf{if}\, i = 0\ \mathsf{then}\ s_0\ \mathsf{else}\ \mathsf{if}\, i = 1\ \mathsf{then}\ s_1\ \mathsf{else} \ldots \mathsf{if}\, i = k-1\ \mathsf{then}\ s_{k-1}\ \mathsf{else}\ \bot_{\mathsf{bool}}$$

où i est une variable de type nat et s_0, …, s_{k-1} et $k-1$ sont les constantes syntaxiques adaptées à s et k.

On a donc, pour tout $k \in \mathbb{N}$, $[\![p/b]\!](f_{s<k}) = [\![p/b\ \ q_{s<k}]\!] = [\![p\ \ b :: q_{s<k}]\!] = [\![p]\!]([\![b :: q_{s<k}]\!])$. Or $[\![b :: q_{s<k}]\!]$ vaut bien sûr $[\![q_{b::s<1+k}]\!] = f_{b::s<1+k}$, d'où le résultat.

3. (a) On note $\pi_0 = p$ et pour tout $\ell < i$, $\pi_{\ell+1} = \pi_\ell / s_\ell$. On veut montrer que $K(\pi_i) = \{0\}$. Soit $s' \in \mathbb{B}^{\mathbb{N}}$ et $k \in \mathbb{N}$. On a, pour tout $\ell < i$, $[\![\pi_{\ell+1}]\!](f_{s'<k}) = [\![\pi_\ell]\!](f_{s_\ell::s'<k+1})$. Par récurrence, on obtient que $[\![\pi_{\ell+1}]\!](f_{s'<k}) = [\![p]\!](f_{s_0::\cdots::s_\ell::s'<\ell+1+k})$ et en particulier

$$[\![\pi_i]\!](f_{s'<k}) = [\![p]\!](f_{s_0::\cdots::s_{i-1}::s'<i+k}),$$

puis

$$[\![\pi_i]\!](f_{s'<0}) = [\![p]\!](f_{s_0::\cdots::s_{i-1}::s'<i}).$$

Or, en revenant à la définition, $f_{s_0::\cdots::s_{i-1}::s'<i} = f_{s<i}$, donc $[\![\pi_i]\!](f_{s'<0}) = \mathsf{uu}$. Ainsi, $k(s', \pi_i) \leq 0$ puis lui est égal. D'où le résultat.

   (b) Supposons $K(p)$ non borné. Soit $A \in \mathbb{N}$: il existe $s \in \mathbb{B}^{\mathbb{N}}$ tel que $k(s,p) \geq A+1$. Soit $b = s_0$ et s' la suite telle que $s = s_0 :: s'$. Pour tout $k \in \mathbb{N}$, $[\![p/b]\!](f_{s'<k}) = [\![p]\!](f_{s<k+1})$. Ainsi, $k(s', p/b) = k(s,p) - 1$, donc $k(s', p/b) \geq A$. D'où le résultat.

   (c) Par l'absurde, supposons $K(p)$ non borné. En itérant le procédé ci-dessus, on construit une suite $s \in \mathbb{B}^{\mathbb{N}}$ et une suite de programmes π telles que $\pi_0 = p$, $\pi_{\ell+1} = \pi_\ell / s_\ell$ et $K(\pi_\ell)$ est non borné.

   On a $[\![p]\!](f_s) = \mathsf{uu}$ par définition de p. Soit donc $i = k(s,p)$. On a vu en (a) que $K(\pi_i) = \{0\}$, qui est borné. Contradiction.

4. (a) Soit $i = k(b::s, p)$. On a $i-1 \in \mathbb{N}$ et $[\![p/b]\!](f_{s<i-1}) = [\![p]\!](f_{b::s<i}) = \mathsf{uu}$, donc, par minimalité, $k(s, p/b) \leq i-1$, d'où le résultat.

(b) Soit $b \in \mathbb{B}$. Remarquons que $K(p/b)$ est une partie bornée non vide de $\mathbb{N}$, donc sa borne supérieure est atteinte. Ainsi, il existe $s \in \mathbb{B}^{\mathbb{N}}$ tel que $k(s, p/b) = |p/b|$. Si $k(b :: s, p) > 0$, alors par la question précédente, $k(s, p/b) < k(b :: s, p)$, donc $|p/b| < k(b :: s, p) \leq |p|$.

Reste à prouver qu'on n'a pas $k(b :: s, p) = 0$. Cela voudrait dire que $[\![p]\!](f_{b::s<0}) = \mathsf{uu}$. Mais $f_{b::s<0}$ est la fonction constante égale à $\bot_{\mathsf{bool}}$; elle ne dépend pas réellement de b et s : par conséquent on aurait $K(p) = \{0\}$ et donc $|p| = 0$ contrairement à l'hypothèse.

5. Soit

$$\forall_{\mathbb{C}} := \mathsf{rec}\,\mathsf{fun}\, f \to \mathsf{fun}\, p \to p\ \ (\mathsf{fun}\, x \to \bot_{\mathsf{bool}})\ ||_{\mathsf{u}}\ (f\ \ (p/0)\ \&_{\mathsf{u}}\ f\ \ (p/1)).$$

On a $[\![\forall_{\mathbb{C}}\ p]\!] = \mathsf{uu}$ ssi $[\![p]\!](f_{s<0}) = \mathsf{uu}$ (pour toute suite s, puisque ça ne dépend pas réellement de s) ou, récursivement, pour tout $b \in \mathbb{B}$ à chaque étape (ce qui permet d'énumérer progressivement toutes les suites s), les $p/b/\ldots$ finissent par vérifier ce critère.

Un programme p qui vérifie ce critère est tel que $[\![p]\!](f) = \mathsf{uu}$ pour tout $f \in \mathbb{C}$. Les questions précédentes ont montré que ce critère est également nécessaire.