Calculs de Processus – EXAMEN

DEA Programmation 2002-2003

17 décembre 2002

I. On spécifie une notion générale $|_\bullet$ de composition parallèle de deux processus de la façon suivante. Soit Act un ensemble d'actions, auquel on adjoint deux éléments spéciaux $\star$ et 0, et une opération $\bullet : (Act \cup \{\star\}) \times (Act \cup \{\star\}) \to (Act \cup \{\star, 0\})$. On spécifie la sémantique opérationnelle comme suit:

$$\frac{P \xrightarrow{\kappa} P' \quad Q \xrightarrow{\lambda} Q' \quad \kappa \bullet \lambda \neq 0}{P \,|_\bullet\, Q \xrightarrow{\kappa\bullet\lambda} P' \,|_\bullet\, Q'}$$

avec la convention que $P \xrightarrow{\star} Q$ signifie qu'aucune transition n'a eu lieu (et donc que $P = Q$).

1. Prenant $Act = L \cup \{\bar{l} \mid l \in L\} \cup \{\tau\}$ (l'ensemble des actions pour CCS), définir $\bullet$ de sorte à retrouver exactement les règles de la sémantique opérationnelle de la composition parallèle de CCS, i.e., définir $\bullet$ tel que $|_\bullet = |$.

2. On se donne maintenant $Act' = \{?l \mid l \in L\} \cup \{!l \mid l \in L\} \cup L$ et l'opération $\bullet$ définie par

$$\begin{array}{l} \forall \kappa \in Act' \cup \{\star\} \quad \star \bullet \kappa = \kappa \bullet \star = \kappa \\ \forall a \in L \;\; (?a) \bullet (!a) = (!a) \bullet (?a) = a \\ \kappa \bullet \lambda = 0 \text{ dans tous les autres cas} \end{array}$$

   On note $\| = |_\bullet$ la composition parallèle ainsi spécifiée, et on se donne le calcul de processus CCS' suivant (avec définitions récursives): $P ::= \Sigma_{i\in I}\kappa_i \cdot P_i \mid (P_1 \parallel P_2) \mid K$, où les κ_i sont des éléments de *Act'*. En quoi ce calcul diffère-t-il du fragment correspondant de CCS?

3. Dessiner l'automate décrivant les transitions (au sens de CCS') de $P_1 \parallel P_2 \parallel P_3$, où:

$$\begin{array}{l} P_1 = ((?a_2) \cdot (!c) \cdot P_1) + ((?a_2) \cdot (!t) \cdot P_1) \\ P_2 = ((?a_1) \cdot (!t) \cdot P_2) + b \cdot 0 \\ P_3 = ((!a_2) \cdot (?c) \cdot P_3) + ((!a_1) \cdot (?t) \cdot 0) \end{array}$$

II. Prouver dans CCS finitaire (c'est-à-dire sans les constantes K) les équations suivantes:

$$P \mid Q \approx Q \mid P \qquad ((\nu a)P) \mid Q \approx (\nu a)\,(P \mid Q) \;\; (a \text{ non libre dans } Q)$$

en utilisant seulement les deux équations suivantes:

$$\begin{array}{c} P \mid Q = \Sigma\{\mu \cdot (P' \mid Q) \mid P \xrightarrow{\mu} P'\} + \Sigma\{\mu \cdot (P \mid Q') \mid Q \xrightarrow{\mu} Q'\} \\ +\Sigma\{\tau \cdot (P' \mid Q' \mid P \xrightarrow{\alpha} P' \text{ et } Q \xrightarrow{\overline{\alpha}} Q'\} \end{array}$$

$$(\nu a)\,(\Sigma_{i\in I}\mu_i \cdot P_i) = \Sigma_{\{} j \in I \mid \mu_j \neq a, \overline{a}\}\mu_j \cdot ((\nu a).P_j)$$

et un raisonnement par récurrence sur la taille des processus.

III. On rappelle la notion d'ensemble de règles. Si X est un ensemble fixé, un ensemble de règles K sur X est un ensemble de paires (Y, x), où $Y \subseteq X$ et $x \in X$. Si K est donné, on définit un opérateur $G_K : \mathcal{P}(X) \to \mathcal{P}(X)$ comme suit:

$$G_K(R) = \{x \in X \mid \exists (Y, x) \in K \;\; Y \subseteq R\} .$$

1. Montrer que G_K est monotone.
2. Trouver un ensemble K' de règles sur X tel que $\mathit{lfp}(G_K)^c = \mathit{gfp}(G_{K'})$ (c'est-à-dire tel que le plus petit point fixe de G_K et le plus grand point fixe de $G_{K'}$ sont complémentaires: $\mathit{lfp}(G_K) \cap \mathit{gfp}(G_{K'}) = \emptyset$ et $\mathit{lfp}(G_K) \cup \mathit{gfp}(G_{K'}) = X$), et prouver cette égalité.
3. Appliquer ceci à la convergence / divergence dans le λ-calcul.

IV. On rappelle l'exemple de la chaîne de cellules: $P\langle h_0\rangle = I\langle h_0\rangle \mid G \mid H\langle h_0\rangle$, où

$$I\langle h\rangle = (\nu i)\ (\bar{i}h \mid !i(z) \cdot n(a) \cdot \overline{z}a \cdot \bar{i}a) \qquad G = !(\nu p)\ \overline{n}p \cdot p(z) \cdot !\overline{p}z \qquad H\langle h\rangle = h(z) \cdot !\overline{h}z$$

Adapter ce codage à la situation où les cellules successives contiennent les éléments d'une suite $0, f(0), f(f(0)), \ldots$, en plus du pointeur sur la cellule suivante. Plus précisément, on suppose que le π-calcul est étendu avec un langage de termes de premier ordre que l'on peut passer comme dans CCS avec valeurs, comme suit: $P ::= \mu \cdot P \mid (P \mid P) \mid (\nu a)P \mid !P$, où

$$\mu ::= a(\vec{x}) \mid a\langle \vec{v}\rangle \qquad v ::= a \mid t \qquad t ::= \underline{x} \mid \underline{0} \mid \underline{f(t)}$$

Et il s'agit de définir $P\langle h, \underline{x}\rangle$ tel que

$$P\langle h_0, \underline{0}\rangle \to^* (\nu h_1, \ldots, h_i)\ (!\overline{h_0}\langle h_1, \underline{0}\rangle \mid !\overline{h_1}\langle h_2, \underline{f(0)}\rangle \mid \ldots \mid !\overline{h_{i-1}}\langle h_i, \underline{f^{i-1}(0)}\rangle \mid P\langle h_i, \underline{f^i(0)}\rangle) .$$

V. Soit S défini comme suit (on suppose que le π-calcul est augmenté d'un test sur l'égalité des noms):

$$\begin{aligned} P\langle a, b, n, s\rangle = a(t, x) \cdot\ & \mathit{if}\ t = s \\ & \mathit{then}\ \overline{x}\langle n\rangle \cdot P\langle a, b, n, s\rangle \\ & \mathit{else\ if}\ b = e \\ & \quad \mathit{then}\ \overline{x}\langle n+1\rangle \cdot ((\nu c)\ \overline{z}\langle c, n+1, t\rangle \cdot P\langle a, c, n, s\rangle) \\ & \quad \mathit{else}\ \overline{b}\langle t, x\rangle \cdot P\langle a, b, n, s\rangle \end{aligned}$$

1. Représenter graphiquement (en supposant s_1, s_2, s_3 distincts) l'exécution de

   $$(!z(a, n, s) \cdot P\langle a, e, n, s\rangle) \mid P\langle a_0, e, 1, s_1\rangle \mid \overline{a_0}\langle s_2, x\rangle \mid \overline{a_0}\langle s_3, x\rangle \mid \overline{a_0}\langle s_2, x\rangle \mid x(n_1) \cdot x(n_2) \cdot x(n_3)$$

   et expliquer la structure de donnée codée par P, ainsi que le rôle de tous les canaux.
2. Quelle modification est induite si l'on remplace dans la dernière ligne $\overline{b}\langle t, x\rangle \cdot P\langle a, b, n, s\rangle$ par $(\nu y)\ \overline{b}\langle t, y\rangle \cdot y(m) \cdot \overline{x}\langle m\rangle \cdot P\langle a, b, n, s\rangle$?