

Calculs de processus

Examen final

21 mars 2017

Durée de l'épreuve : 3 heures.

Tous les documents sont autorisés. Le barème est marqué dans chaque exercice.

Exercice 1 (6 points). On considère le langage Imp_{\parallel} introduit en cours. Soient :

$$\begin{aligned} S &:= x := y \\ P &:= z := y; x := z \\ Q &:= \text{new } z = 0 \text{ in } (z := y; x := z) \end{aligned}$$

Déterminer si les équivalences suivantes sont vraies ou fausses :

1. (3 points) $S \simeq_{IO}^c P$?
2. (3 points) $S \simeq_{IO}^c Q$?

Exercice 2 (9 points). On considère les trois processus de CCS suivants :

$$\begin{aligned} S &:= a.b \\ P &:= a \mid \tau.b \\ Q &:= \nu c(a.\bar{c} \mid c.b) \\ R &:= \nu c(a.R_0 \mid c.b) \end{aligned} \quad \text{où } R_0 \stackrel{\text{rec}}{=} \tau.(R_0 \mid \bar{c}).$$

Déterminer si les bisimilarités suivantes sont vraies ou fausses :

1. (3 points) $P \approx S$?
2. (3 points) $Q \approx S$?
3. (3 points) $R \approx S$?

Exercice 3 (5 points). Soit

$$H \stackrel{\text{rec}}{=} r.b.H + g.b.H$$

Trouver un processus P de CCS de la forme

$$\nu(\tilde{c})(K_r \mid K_g \mid K_b \mid V)$$

où K_r ne contient pas b, g , K_g ne contient pas r, b , K_b ne contient pas r, g et V ne contient pas r, g, b , tel que $P \approx H$.

Corrigé

Exercice 1.

1. On a $S \not\approx_{IO}^c P$ (on a vu cela en TD, c'était dans le sujet d'examen de l'année passée, avec des noms de variables légèrement différents). Un contexte qui les sépare est par exemple

$$C := [\cdot] \mid z := 0.$$

En effet, si s est une mémoire dans laquelle y vaut $n \neq 0$, $(C[S], s)$ a deux exécutions qui terminent (selon qui entre $x := y$ et $z := 0$ est exécuté en premier), les deux terminant avec la mémoire $s[x \mapsto n, z \mapsto 0]$; en particulier, la valeur finale de x est différente de 0. En revanche, $(C[P], s)$ a une exécution terminante de la forme

$$\begin{aligned} (C[P], s) &\rightarrow^* (x := z \mid z := 0, s[z \mapsto n]) \rightarrow (x := z \mid \text{skip}, s[z \mapsto 0]) \\ &\rightarrow (\text{skip} \mid \text{skip}, s[x \mapsto 0, z \mapsto 0]) \end{aligned}$$

où la valeur finale de x est 0. Cela montre que $\llbracket C[P] \rrbracket^{IO} \neq \llbracket C[S] \rrbracket^{IO}$, car $(s, s[x \mapsto 0, z \mapsto 0]) \in \llbracket C[P] \rrbracket^{IO} \setminus \llbracket C[S] \rrbracket^{IO}$.

2. On a bien $S \simeq_{IO}^c Q$, car $\llbracket S \rrbracket^{TE} = \llbracket Q \rrbracket^{TE}$. En effet, les traces de $\llbracket S \rrbracket^{TE}$ ont toutes la forme

$$(s_1, s_1) \cdots (s_n, s_n)(s_{n+1}, s_{n+1}[x \mapsto k])(s_{n+2}, s_{n+2}) \cdots (s_m, s_m)$$

où k est la valeur de y dans s_{n+1} . Dans Q , l'instruction $z := y$ n'est pas visible de l'extérieur, car z est locale : on a

$$(Q, s) \rightarrow (\text{new } z = k \text{ in } (\text{skip}; x := z), s) \rightarrow (\text{new } z = k \text{ in } x := z, s),$$

c'est-à-dire, après l'exécution de l'affectation $z := y$ (on suppose que y a la valeur k dans s) et la suppression du `skip` qui en résulte, la variable locale z a la valeur k , mais cette valeur n'est pas visible de l'extérieur, la mémoire reste égale à s . L'exécution de cette affectation est donc assimilable à l'exécution d'un `skip`. Après, comme z est locale, personne ne peut en modifier la valeur, donc l'affectation $x := z$ a pour effet de donner la valeur k à x , exactement comme dans l'instruction $x := y$ (car k est la valeur de y dans s). Les traces de $\llbracket Q \rrbracket^{TE}$ ont donc la même forme que celles de $\llbracket S \rrbracket^{TE}$.

Exercice 2.

1. On a $P \not\approx S$. En effet, l'opposant a une stratégie gagnante : il suffit qu'il joue le τ dans P . Voici la vérification du fait que cette stratégie est gagnante :

$$O : P \xrightarrow{\tau} a \mid b$$

J : obligé de rester sur S (S n'a pas de transition τ)

$$O : a \mid b \xrightarrow{b} a \mid \mathbf{0}$$

J : perdu! (S n'a pas de transition b)

2. On a $Q \approx S$. Voici une bisimulation, en utilisant les couleurs pour le couplage :

$$S \xrightarrow{a} b \xrightarrow{b} \mathbf{0} \qquad Q \xrightarrow{a} \nu c(\bar{c} \mid c.b) \xrightarrow{\tau} b \xrightarrow{b} \mathbf{0}$$

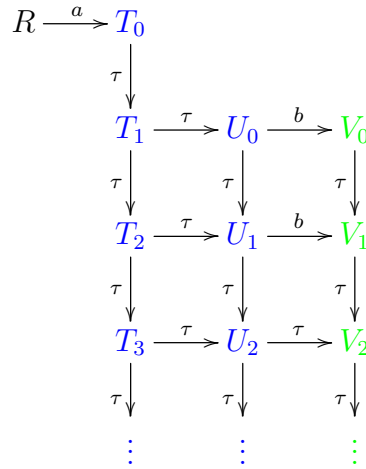
3. On a $R \approx S$. Écrivons d'abord le système de transitions étiquetées de R . Comme il est infini, il vaut mieux de commencer par fixer des notations :

$$T_n := \nu c(R_0 \mid \overbrace{\bar{c} \mid \cdots \mid \bar{c}}^n \mid c.b)$$

$$U_n := \nu c(R_0 \mid \overbrace{\bar{c} \mid \cdots \mid \bar{c}}^n \mid b)$$

$$V_n := \nu c(R_0 \mid \overbrace{\bar{c} \mid \cdots \mid \bar{c}}^n)$$

Voici donc le système de transitions étiquetées de R , les couleurs indiquant le couplage avec celui de S pour la bisimulation :



Exercice 3. Voici une solution possible :

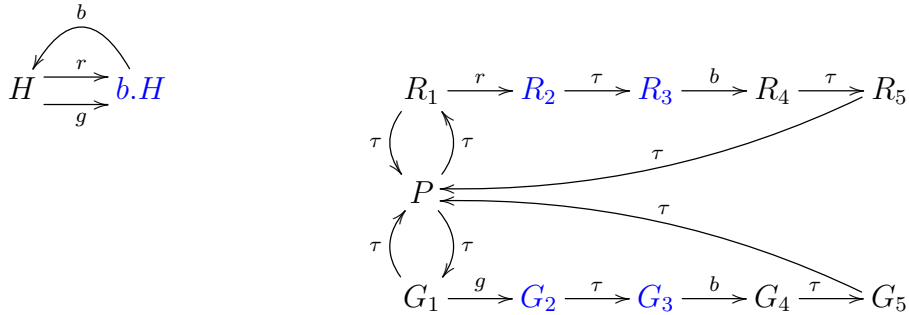
$$K_r \stackrel{\text{rec}}{=} \bar{p}.(\bar{v}.K_r + r.\bar{c}.c.\bar{v}.K_r)$$

$$K_g \stackrel{\text{rec}}{=} \bar{p}.(\bar{v}.K_g + g.\bar{c}.c.\bar{v}.K_g)$$

$$K_b \stackrel{\text{rec}}{=} c.b.\bar{c}.K_b$$

$$V \stackrel{\text{rec}}{=} p.v.V$$

avec $\tilde{c} = p, v, c$. On reconnaîtra que V n'est rien d'autre qu'un verrou. La bisimulation est donné par le couplage suivant (indiqué par les couleurs) :



Il n'est pas nécessaire de spécifier les processus R_1, \dots, R_5 et G_1, \dots, G_5 (on peut les retrouver facilement à partir des transitions). Cependant, les voici pour complétude :

$$\begin{array}{ll}
R_1 := \nu\tilde{c}(\bar{v}.K_r + r.\bar{c}.c.\bar{v}.K_r \mid K_g \mid K_b \mid v.V) & G_1 := \nu\tilde{c}(K_r \mid \bar{v}.K_g + g.\bar{c}.c.\bar{v}.K_g \mid K_b \mid v.V) \\
R_2 := \nu\tilde{c}(\bar{c}.c.\bar{v}.K_r \mid K_g \mid K_b \mid v.V) & G_2 := \nu\tilde{c}(K_r \mid \bar{c}.c.\bar{v}.K_g \mid K_b \mid v.V) \\
R_3 := \nu\tilde{c}(c.\bar{v}.K_r \mid K_g \mid b.\bar{c}.K_b \mid v.V) & G_3 := \nu\tilde{c}(K_r \mid c.\bar{v}.K_g \mid b.\bar{c}.K_b \mid v.V) \\
R_4 := \nu\tilde{c}(c.\bar{v}.K_r \mid K_g \mid \bar{c}.K_b \mid v.V) & G_4 := \nu\tilde{c}(K_r \mid c.\bar{v}.K_g \mid \bar{c}.K_b \mid v.V) \\
R_5 := \nu\tilde{c}(\bar{v}.K_r \mid K_g \mid K_b \mid v.V) & G_5 := \nu\tilde{c}(K_r \mid \bar{v}.K_g \mid K_b \mid v.V)
\end{array}$$