Calculs de processus
Examen final

16 décembre 2015

Durée de l'épreuve : 3 heures.
Tous les documents sont autorisés.
Le barème est marqué dans chaque exercice. Les exercices 4 et 5 sont des « bonus ». Sans bonus, on arrive à 19/20. Avec les bonus, on peut dépasser 20/20 (en effet, on arrive à 24/20).

Exercice 1 (9 points). On considère le langage $Imp_{\|}$ introduit en cours et, pour tout processus P ne contenant ni de while ni de await, on définit

$$\mathsf{atomic}\, P \;:=\; \mathsf{await}\, \mathsf{true}\, \mathsf{do}\, P.$$

Soient :

$$\begin{aligned} S :=&\quad x := z \\ P :=&\quad y := z;\, x := y \\ Q :=&\quad x := z;\, z := z \\ R :=&\quad \mathsf{new}\, y = 0\, \mathsf{in}\, \mathsf{atomic}\, (y := z;\, x := y) \end{aligned}$$

Déterminer si les équivalences suivantes sont vraies ou fausses :

1. (**3 points**) $S \simeq^{c}_{IO} P$?
2. (**3 points**) $S \simeq^{c}_{IO} Q$?
3. (**3 points**) $S \simeq^{c}_{IO} R$?

Exercice 2 (6 points). Déterminer si les bisimilarités suivantes sont vraies ou fausses :

1. (**3 points**) $\tau.a \mid b \overset{?}{\approx} \tau.(a \mid b)$;
2. (**3 points**) $\tau.a + b \overset{?}{\approx} P$, où P est défini de la manière suivante :

$$\begin{aligned} P &\overset{\text{rec}}{=} a + \tau.Q \\ Q &\overset{\text{rec}}{=} b + \tau.P \end{aligned}$$

Exercice 3 (4 points). Soit

$$H \stackrel{\text{rec}}{=} b.a.b.a.a.H$$

Trouver un processus P de CCS de la forme

$$\nu(\widetilde{c})(K_a \mid K_b)$$

où K_a ne contient pas b et K_b ne contient pas a, tel que $P \approx H$.

Exercice 4 (2 points). Soit $R := a.a.(b.c + c.b)$. Montrer que $R \sim P \mid Q$ implique $P \sim \mathbf{0}$ ou $Q \sim \mathbf{0}$.

Exercice 5 (3 points). Soit $H_0, H_1, H_2, \ldots$ une famille de processus définis par les équations récursives suivantes (une pour chaque $n \in \mathbb{N}$) :

$$H_n \stackrel{\text{rec}}{=} \overbrace{\overline{a}.\ldots.\overline{a}}^{n}.b.H_{n+1}.$$

Trouver un processus P de CCS défini à l'aide d'un nombre *fini* d'équations récursives tel que $P \approx H_0$.

Corrigé

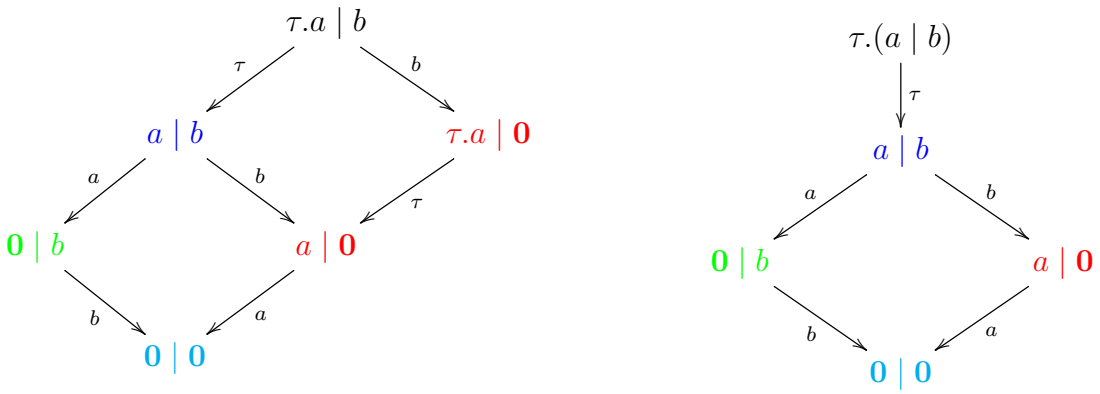
Exercice 1. Écrivons tout d'abord la sémantique de S :

$$[\![S]\!]^{TE} = c(\{(s, s[s(z)/x])\}).$$

1. On a $S \not\simeq^c_{IO} P$. Pour le montrer, il suffit de remarquer que la sémantique de P est égale à
$$[\![P]\!]^{TE} = \{\alpha\beta \mid \alpha \in [\![y := z]\!]^{TE},\ \beta \in [\![x := y]\!]^{TE}\}$$
et donc, si on pose $\gamma := (s, s[s(z)/y])(s', s'[s'(y)/x])$, on a $\gamma \in [\![P]\!]^{TE}$ tandis que $\gamma \notin [\![S]\!]^{TE}$ pour un choix approprié de s, s'. On peut également vérifier que
$$C \ := \ [\cdot] \mid y := 0$$
est un contexte qui sépare S et P.
2. On a $S \simeq^c_{IO} Q$. En effet, le processus $z := z$ ne modifie pas la mémoire : sa seule exécution est $(z := z, s) \to (\mathsf{skip}, s)$ (car $s[s(z)/z] = s$), ce qui implique
$$[\![z := z]\!]^{TE} = c(\{(s, s) \mid s \in \mathrm{St}\}) = [\![\mathsf{skip}]\!]^{TE}$$
et donc $(z := z) \simeq^c_{IO} \mathsf{skip}$. Mais skip est l'élement neutre de la composition séquentielle et donc
$$Q = (S;\ z := z) \simeq^c_{IO} (S; \mathsf{skip}) \simeq^c_{IO} S.$$
3. On a $S \simeq^c_{IO} R$. En effet, on remarque que
$$R = \quad \mathsf{new}\, y = 0 \,\mathsf{in}\, \mathsf{atomic}\, P$$
où P est le processus du point 1. Or, exécuté de manière atomique, P a le même comportement que S, mise à part la modification de y :
$$[\![\mathsf{atomic}\, P]\!]^{TE} = c(\{(s, s[s(z)/x][s(z)/y]) \mid s \in \mathrm{St}\}).$$
Mais dans R, y est une variable locale, donc $[\![R]\!]^{TE} = [\![S]\!]^{TE}$.

Exercice 2.

1. Les deux processus sont bien bisimilaires. Une bisimulation qui les contient est la suivante (le couplage de la bisimulation est donné par les couleurs) :



2. Les deux processus ne sont pas bisimilaires. Posons $R := \tau.a + b$. Une stratégie gagnante pour l'opposant peut être définie comme suit :

O :	$R \xrightarrow{\tau} a$	
J :	$P \Longrightarrow P$	$P \Longrightarrow Q$
O :	$P \xrightarrow{\tau} Q$	$Q \xrightarrow{b} \mathbf{0}$
J :	obligé de rester sur a	perdu !
O :	$Q \xrightarrow{b} \mathbf{0}$	
J :	perdu !	

(Explication : après le premier coup de l'opposant, le joueur a le droit d'effectuer un nombre quelconque de transitions τ à partir de P ; la première colonne correspond à un nombre pair de transitions, la deuxième colonne à un nombre impair)

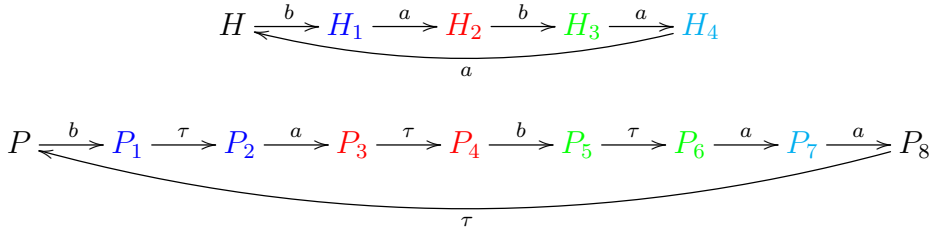
Exercice 3. Soient

$$K_a \overset{\text{rec}}{=} c.a.\bar{c}.c.a.a.\bar{c}.K_a$$
$$K_b \overset{\text{rec}}{=} b.\bar{c}.c.K_b$$

La définition

$$P := \nu c(K_a \mid K_b)$$

est une solution à la question. En effet, on a bien $P \approx H$, comme le montre la bisimulation suivante (le couplage est donné par les couleurs) :



Exercice 4. On fait la preuve par contraposition : on montre que $P \not\sim \mathbf{0}$ et $Q \not\sim \mathbf{0}$ implique $R \not\sim P \mid Q$. On commence par remarquer qu'un processus est fortement bisimilaire à $\mathbf{0}$ ssi il n'a aucune transition, donc $P \not\sim \mathbf{0}$ et $Q \not\sim \mathbf{0}$ équivaut au fait qu'il existe deux transitions $P \xrightarrow{\alpha} P'$ et $Q \xrightarrow{\beta} Q'$. Si l'une entre α, β est différente de a, on a immédiatement absence de bisimulation forte (car la seule transition possible pour R est $R \xrightarrow{a} a.(b.c + c.b)$). Donc $P \xrightarrow{a} P'$ et $Q \xrightarrow{a} Q'$. On commence alors le jeu de bisimulation ainsi :

O :	$P \mid Q \xrightarrow{a} P' \mid Q$	
J :	$R \xrightarrow{a} a.(b.c + c.b)$	(obligé)
O :	$P' \mid Q \xrightarrow{a} P' \mid Q'$	
J :	$a.(b.c + c.b) \xrightarrow{a} b.c + c.b$	(obligé)

Maintenant, si ni P' ni Q' est capable d'une transition b, le joueur perd immédiatement :

O :	$b.c + c.b \xrightarrow{b} c$
J :	perdu !

Supposons donc que $P' \xrightarrow{b} P''$ (on pourrait aussi supposer que $Q' \xrightarrow{b} Q''$; on invite le lecteur à vérifier que ce cas est strictement analogue et conduit à la même conclusion). On considère alors une autre partie, qui commence de la même façon :

O : $P \mid Q \xrightarrow{a} P' \mid Q$
J : $R \xrightarrow{a} a.(b.c + c.b)$ (obligé)
O : $P' \mid Q \xrightarrow{b} P'' \mid Q'$
J : perdu !

Dans tous les cas, l'opposant a une stratégie gagnante, donc $R \not\sim P \mid Q$.

Exercice 5. Voilà une solution possible, avec deux équations récursives :

$$C(x) \overset{\text{rec}}{=} u.\nu c(x.a.\overline{u}'.\overline{c} \mid C(c)) + x.b.\overline{u}'.(\overline{c}'_0 \mid C(c_0))$$
$$C'(x) \overset{\text{rec}}{=} u'.\nu c(x.a.\overline{u}.\overline{c} \mid C'(c)) + x.b.\overline{u}.(\overline{c}_0 \mid C'(c'_0))$$
$$P := \nu(u, u', c_0, c'_0)(\overline{c}_0 \mid C(c_0) \mid C'(c'_0))$$

L'idée est la suivante : $C(c_0)$ et $C'(c'_0)$ sont deux compteurs parfaitement symétriques, initialisés à 0 ; C recoit ses signaux d'augmentation sur u, C' sur u'. La forme générale du compteur C, quand il contient la valeur n, est la suivante (celle du compteur C' est obtenue en remplaçant u par u' et c_i par c'_i) :

$$\nu(c_1, \ldots, c_n)(c_0.a.\overline{u}'.\overline{c}_1 \mid \cdots \mid c_{n-1}.a.\overline{u}'.\overline{c}_n \mid C(c_n))$$

Un tel compteur est capable de deux actions :

— recevoir un signal d'augmentation sur u et évoluer vers l'état $n+1$;
— recevoir un signal de réinitialisation sur c_0, qui déclenche une sorte de réaction en chaîne : une action a, suivie par un signal d'augmentation de C', une synchronisation sur c_1 et ainsi de suite jusqu'à la dernière syncronisation sur c_n (qui se fait avec $C(c_n)$), ce qui déclenche une action b, un autre signal d'augmentation de C' et une émission sur c'_0 en parallèle avec $C(c_0)$ (c'est-à-dire, l'état initial).

Après la réinisialitation de C, C' a reçu $n+1$ signaux d'augmentation. Si l'on suppose qu'il était dans l'état 0, il se trouve maintenant dans l'état $n+1$, alors que C est dans l'état 0. La réaction en chaîne donc comporte la diminution de C à faveur de l'augmentation de C'.

A ce moment, l'émission sur c'_0 déclenche une réaction en chaîne symétrique, qui voit C' envoyer $n+2$ signaux d'augmentation vers C, en même temps que $n+1$ actions a et une action b, et le processus continue.

Schématiquement, les réduits du processus P ont toujours deux composantes parallèles, que l'on peut appeler C_n et $C'_{n'}$, correspondant au fait que le compteur C est dans l'état n et le compteur C' est dans l'état n'. L'une des ces deux composantes est « active » : il s'agit du compteur qui en train de décrementer. Les actions possibles sont les suivantes :

$$C_{n+1} \mid C'_{n'} \overset{a}{\Longrightarrow} C_n \mid C'_{n'+1} \qquad C_n \mid C_{n'+1} \overset{a}{\Longrightarrow} C_{n+1} \mid C'_{n'}$$
$$C_0 \mid C'_{n'} \overset{b}{\Longrightarrow} C_0 \mid C'_{n'+1} \qquad C_n \mid C'_0 \overset{b}{\Longrightarrow} C_{n+1} \mid C'_0$$

où l'on a mis en évidence en bleu la composante active. Le processus P est initialement dans l'état $C_0 \mid C'_0$. Il est maintenant clair qu'il existe une bisimulation entre P et H_0.

Voici les premières transitions de P :

$$C_0 \mid C'_0 \overset{b}{\Longrightarrow} C_0 \mid C'_1 \overset{a}{\Longrightarrow} C_1 \mid C'_0 \overset{b}{\Longrightarrow} C_2 \mid C'_0 \overset{a}{\Longrightarrow} C_1 \mid C'_1 \overset{a}{\Longrightarrow} C_0 \mid C'_2 \overset{b}{\Longrightarrow} C_0 \mid C_3 \overset{a}{\Longrightarrow} \cdots$$

On observe le « ping-pong » entre les deux compteurs.

On remarque aussi que, les deux compteurs étant complèment symétriques, il est possible d'inclure u, c_0, u', c'_0 dans les paramètres de la définition récursive, de manière à ce que C et C' ne soient que des instances d'un seul compteur. Cela permet de définir un processus bisimilaire à H_0 à l'aide d'une seule équation récursive. Nous avons préféré la solution avec deux équations car la définition récursive avec un seul paramètre est légèrement plus simple.