Examen

DEA

18 décembre 2003

On attachera une grande importance à la concision, à la clarté, et à la précision de la rédaction.

Bisimulations

Question 1 Parmi les relations de bisimulations ($\sim$, $\approx$ et $\cong$), trouver la relation la plus précise reliant :

a) P et $\tau.P$ b) P et $P \mid 0$ c) $(P \mid Q) + (P \mid R)$ et $P \mid (Q + R)$

d) $P + \tau.(P + Q)$ et $\tau.(P + Q)$ e) $\tau.P \mid Q \mid R$ et $P \mid Q \mid R$

(Dans chaque cas, donner une preuve ou un contrexemple si aucune relation n'est vraie)

Corrigé

a) $P \approx \tau.P$. En effet $P \xrightarrow{\alpha} P'$ implique $\tau.P \xrightarrow{\tau}\xrightarrow{\alpha} P'$. Donc $P \stackrel{\alpha}{\Longrightarrow} P'$. Réciproquement, $\tau.P \xrightarrow{\tau} P$ et $P \stackrel{\widehat{\tau}}{\Longrightarrow} P$.

b) $P \sim P \mid 0$, puisque $P \xrightarrow{\alpha} P'$ si et seulement si $P \mid 0 \xrightarrow{\alpha} P' \mid 0$. En considérant les paires $(P, P \mid 0)$ on vérifie aisément que c'est une bisimulation forte.

c) Aucune relation ne relie $(P \mid Q) + (P \mid R)$ et $P \mid (Q + R)$. Prendre $R = 0$. On a alors $P \mid Q + P \mid 0 \sim P \mid Q + P$ et $P \mid (Q + R) \sim P \mid Q$. En posant $P = a.0$ et $Q = b.0$ on a bien $a.b.0 + b.a.0 + a.0$ différent de $a.b.0 + b.a.0$.

d) $P + \tau.(P + Q) \cong \tau.(P + Q)$. Si $P + \tau.(P + Q) \xrightarrow{\alpha} P'$ où $P \xrightarrow{\alpha} P'$. Alors $\tau.(P + Q) \stackrel{\alpha}{\Longrightarrow} P'$. Si $P + \tau.(P + Q) \xrightarrow{\tau} P + Q$, alors $\tau.(P + Q) \xrightarrow{\tau} P + Q$. Réciproquement, si $\tau.(P + Q) \xrightarrow{\tau} P + Q$, alors $P + \tau.(P + Q) \xrightarrow{\tau} P + Q$.

e) $\tau.P \mid Q \mid R \approx P \mid Q \mid R$. En fait, comme vu dans le cours, on a plus généralement $\tau.P \mid Q \approx P \mid Q$. On applique la définition sur la bisimulation $\{(\tau.P \mid Q, P \mid Q)\}$.

Note : On devrait également prouver dans chaque cas que la relation est la plus faible. Nous n'avons appliqué aucune pénalité pour manquer la présente partie des questions. Plutôt, nous avons donné un bonus à ces étudiants qui ont résolu l'exercice entièrement.

Terminaison

On dira qu'un processus P *termine* s'il ne possède pas de dérivation infinie $P \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} P_2 \xrightarrow{\alpha_3} P_3 \cdots$ où α_i est une action quelconque, possiblement τ, pour tout $i > 0$.

Question 2 Donner en CCS et en pi-calcul deux exemples de processus qui ne terminent pas.

Corrigé K où $K \stackrel{\text{def}}{=} \tau.K$. Dans le pi-calcul, $!x$ $|!\overline{x}$.

Question 3 En CCS, est-ce que la bisimulation forte ($\sim$), la bisimulation faible ($\approx$), ou la congruence observationnelle ($\cong$) préserve la terminaison. Autrement dit, si P termine et :

a) si $P \sim Q$, alors Q termine ? b) si $P \approx Q$, alors Q termine ? c) si $P \cong Q$, alors Q termine ?

Corrigé

a) Oui. En fait, si Q ne termine pas, alors $Q \xrightarrow{\alpha_1} Q_1 \xrightarrow{\alpha_2} Q_2 \xrightarrow{\alpha_3} Q_3 \ldots$. Par définition de bisimulation forte, si $P \sim Q$ on a aussi $P \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} P_2 \xrightarrow{\alpha_3} P_3 \ldots$. Donc P ne termine pas non plus.

b) Non. En fait, on a $a.0 \approx K$ où $K \stackrel{\text{def}}{=} \tau.K + a.0$. Or le premier ne termine pas, alors que le deuxième termine.

c) Non. En fait, on a $\tau.a.0 \cong K$ où $K \stackrel{\text{def}}{=} \tau.K + \tau.a.0$. Idem.

Un processus P' dérivé de P est tel que $P \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} P_2 \cdots \xrightarrow{\alpha_n} P_n = P'$ ($n \geq 0$). Nous dirons qu'un processus P *termine en* `fin` si, pour tout processus P' dérivé de P, on n'a pas $P' \xrightarrow{\texttt{fin}} P''$, mais on a $P' \sim \overline{\texttt{fin}}.0$ si $P' \xrightarrow{\overline{\texttt{fin}}} P''$.

Question 4 Supposons que P termine en `fin`. Montrer que

a) si $P \xrightarrow{\alpha} P'$, alors P' termine en `fin` ;

b) si $P \sim Q$, alors Q termine en `fin`.

Corrigé

a) $P \xrightarrow{\alpha} P'$. Alors P' termine en `fin` puisque tout dérivé de P' est aussi un dérivé de P.

b) Si $P \sim Q$ et P termine en `fin`. S'il y a un dérivé Q' de Q tel que $Q' \xrightarrow{\texttt{fin}} Q''$. Alors par récurrence sur la longueur de la dérivation de Q à Q', on montre que P a aussi un dérivé P' tel que $P' \xrightarrow{\texttt{fin}} P''$. C'est donc impossible. Si un dérivé Q' de Q vérifie $Q' \xrightarrow{\overline{\texttt{fin}}} Q''$. Alors à nouveau par récurrence sur la longueur de la dérivation de Q à Q', on montre qu'un dérivé de P est tel que $P' \xrightarrow{\overline{\texttt{fin}}} P''$ avec $P' \sim Q'$. Or $Q' \sim \overline{\texttt{fin}}.0$. Par transitivité, $P' \sim \overline{\texttt{fin}}.0$.

Posons $P;Q = (\nu a)(P\{a/\texttt{fin}\} \mid a.Q)$ où a est un nom frais n'apparaissant ni dans P ni dans Q, et où $P\{a/\texttt{fin}\}$ désigne le processus obtenu après la substitution de `fin` par a dans P.

Question 5 Montrer que si P et Q terminent en `fin`, alors $P;Q$ termine aussi en `fin`.

Corrigé

Si $P;Q \xrightarrow{*} R \xrightarrow{\texttt{fin}} R'$, alors comme `fin` est substitué par a, un nom frais dans $P\{a/\texttt{fin}\}$ dans $P;Q$, on ne peut qu'avoir $P \xrightarrow{*} P' \xrightarrow{\overline{\texttt{fin}}}, Q \xrightarrow{*} S \xrightarrow{\texttt{fin}} S'$. C'est impossible puisque Q termine en `fin`.

Si $P;Q \xrightarrow{*} R \xrightarrow{\overline{\texttt{fin}}} R'$, à nouveau on a $P \xrightarrow{*} P' \xrightarrow{\overline{\texttt{fin}}} P'', Q \xrightarrow{*} S \xrightarrow{\overline{\texttt{fin}}} S'$. Donc $P;Q \xrightarrow{*} (\nu a)(P' \mid Q) \xrightarrow{\tau} (\nu a)(P'' \mid Q) \xrightarrow{*} R = (\nu a)(P'' \mid S) \xrightarrow{\overline{\texttt{fin}}} (\nu a)(P'' \mid S')$ with $P' \sim S \sim \overline{\texttt{fin}}.0$. Donc $P'' \sim 0$ et $R = (\nu a)(P'' \mid S) \sim (\nu a)(0 \mid S) \sim (\nu a)S \sim S \sim \overline{\texttt{fin}}.0$ puisque a n'apparait pas dans S.

Question 6 Quelles sont les relations entre :

a) $P;(Q;R)$ et $(P;Q);R$? b) $\overline{\texttt{fin}}.0;P$ et P ? c) $P;\overline{\texttt{fin}}.0$ et P ?

Faut-il supposer que P, Q, R terminent en `fin` ?

Corrigé

a) On montre que les paires $\{(P;(Q;R),(P;Q);R),(P,P)\}$ forment une bisimulation forte.

Si $P;(Q;R) \xrightarrow{\alpha} S$ avec $\alpha \neq \texttt{fin}$ et $\alpha \neq \overline{\texttt{fin}}$. Alors $S = P';(Q;R)$ où $P \xrightarrow{\alpha} P'$. (petit lemme à montrer). Donc $(P;Q);R \xrightarrow{\alpha} (P';Q);R$.

Si $P;(Q;R) \xrightarrow{\tau} \sim Q;R$ car $P \xrightarrow{\overline{\texttt{fin}}} P'$ et donc $P \sim \overline{\texttt{fin}}.0$. Alors $P;Q \xrightarrow{\tau} \sim Q$ et donc $(P;Q);R \xrightarrow{\tau} \sim Q;R$.

Et réciproquement.

b) $\overline{\texttt{fin}}.0;P = (\nu a)(\overline{a}.0 \mid a.P) \xrightarrow{\tau} (\nu a)(0 \mid P) \sim 0 \mid P \sim P$. Donc $\overline{\texttt{fin}}.0;P \approx P$.

c) On montre $P;\overline{\texttt{fin}}.0 \approx P$. En effet, si $P;\overline{\texttt{fin}}.0 \xrightarrow{\alpha} P';\overline{\texttt{fin}}.0$ avec $P \xrightarrow{\alpha} P'$ où $a \neq \texttt{fin}$ et $a \neq \overline{\texttt{fin}}$, alors par co-induction $P';\overline{\texttt{fin}}.0 \approx P'$. Si $P \xrightarrow{\overline{\texttt{fin}}} P'$, alors $P \sim \overline{\texttt{fin}}.0$ et $P;\overline{\texttt{fin}}.0 \sim \tau.\overline{\texttt{fin}}.0$. Donc $P \approx \overline{\texttt{fin}}.0$.

Remarquons que pour montrer (a) et (b) on ne sert pas de l'hypothèse P termine en `fin`. Au contrarie, cette hypothèse est necessaire pour (c).

Diffusion

Le pi-calcul avec *diffusion*, π_δ, est un calcul pour représenter une forme de communication où un message envoyé n'est pas consommé par le récepteur. Après réception, le message est encore disponible pour d'autres processus.

On définit π_δ comme le pi-calcul standard π, sauf que le préfixe $\bar{x}y.P$ est remplacée par $\delta(x,y).P$, avec la règle de sémantique opérationnelle suivante :

$$\delta(x,y).P \xrightarrow{\bar{x}y} \delta(x,y).0 \,|\, P$$

Question 7 Prouver qu'il est possible de coder π_δ dans π avec un codage $[\![\cdot]\!] : \pi_\delta \to \pi$ tel que, pour tout P, on a $[\![P]\!] \sim P$, où $\sim$ est la *late strong bisimulation*. Montrer que cette propriété de bisimulation est en effet satisfaite.

Corrigé Let $[\![\cdot]\!] : \pi_\delta \to \pi$ be defined as follows :

$$[\![\delta(x,y).P]\!] \;=\; \overline{x}y.((!\,\overline{x}y.0) \mid [\![P]\!])$$

and $[\![\cdot]\!]$ homomorphic on all the other operators, namely :

$$\begin{array}{rcl}
[\![0]\!] & = & 0 \\
[\![x(y).P]\!] & = & x(y).[\![P]\!] \\
[\![\tau.P]\!] & = & \tau.[\![P]\!] \\
[\![P \mid Q]\!] & = & [\![P]\!] \mid [\![Q]\!] \\
[\![P + Q]\!] & = & [\![P]\!] + [\![Q]\!] \\
[\![(\nu x)P]\!] & = & (\nu x)[\![P]\!] \\
[\![!\,P]\!] & = & !\,[\![P]\!]
\end{array}$$

We show now that $[\![P]\!] \sim P$. The interesting case is $[\![\delta(x,y).P]\!]$. Observe that only possible transition from $[\![\delta(x,y).P]\!] = \overline{x}y.((!\,\overline{x}y.0) \mid [\![P]\!])$ is

$$\overline{x}y.((!\,\overline{x}y.0) \mid [\![P]\!]) \xrightarrow{\bar{x}y} (!\,\overline{x}y.0) \mid [\![P]\!],$$

while the only possible transition from $\delta(x,y).P$ is

$$\delta(x,y).P \xrightarrow{\bar{x}y} \delta(x,y).0 \,|\, P.$$

It is then sufficient to show that

$$(!\,\overline{x}y.0) \mid [\![P]\!] \;\sim\; [\![\delta(x,y).0 \,|\, P]\!]$$

and then apply transitivity. The above is almost immediate, in fact :

$$\begin{array}{rcl}
[\![\delta(x,y).0 \,|\, P]\!] & = & [\![\delta(x,y).0]\!] \,|\, [\![P]\!] \;\text{(by definition of } [\![\cdot]\!]) \\
& = & \overline{x}y.((!\,\overline{x}y.0) \mid [\![0]\!]) \,|\, [\![P]\!] \;\text{(by definition of } [\![\cdot]\!]) \\
& = & \overline{x}y.((!\,\overline{x}y.0) \mid 0) \,|\, [\![P]\!] \;\text{(by definition of } [\![\cdot]\!]) \\
& \equiv & (\overline{x}y.!\,\overline{x}y.0) \,|\, [\![P]\!] \;\text{(by definition of } \equiv) \\
& \sim & (!\,\overline{x}y.0) \,|\, [\![P]\!] \;\text{(easy lemma).}
\end{array}$$

Question 8 (Bonus) Discuter s'il existe un codage dans l'autre direction (avec des propriétés "raisonnables") ou pas. Dans le cas positif, définir le codage et discuter ses propriétés sémantiques (la preuve n'est pas exigée). Dans le cas négatif, expliquer pourquoi ce n'est pas possible.

Corrigé The existence of the inverse encoding depends on the properties we require it to satisfy (as it is usually the case).

An encoding correct wrt bisimulation (even weak bisimulation) does not exist, because in π_δ any process that can do a transition labeled by an output action has at least one derivation where the same transition is repeated infinitely many times.

However, if we consider a weaker semantics condition, for instance the preservation of the equivalence induced by weak barbed bisimulation, then an encoding can be given. This condition means that for arbitrary P and Q in π, $[\![P]\!] \dot{\approx} [\![Q]\!]$ if and only if $P \dot{\approx} Q$, where $\dot{\approx}$ is weak barbed bisimilarity. We discuss one such possible encoding below. For simplicity we restrict to the asynchronous pi-calculus and we use recursion instead of replication.

Let $[\![\cdot]\!]$ be homomorphic except for senders and receivers, for which it is defined as follows :

$$[\![\bar{x}y]\!] = (\nu s)(!\,\bar{x}s \,|\, s(r\,t\,f).(!\,\bar{r}t \,|\, !\,\bar{t}y \,|\, !\,s(r\,t\,f).!\,t.!\,\bar{r}f)) \quad (1)$$

$$[\![x(y)P]\!] = A, \text{ where } A \stackrel{def}{=} x(s).(\nu r)(\nu t)(\nu f)(!\,\bar{s}\,r\,t\,f \,|\, r(z).(!\,\bar{t} \,|\, [z=t]t(y).P \,|\, [z=f]A)) \quad (2)$$

Intuitively, a sender behaves as follows :

1. It creates a secret channel, s.
2. It sends s on x, thereby indicating that it is ready to communicate. This message may be received by zero or more receivers (at least one if there is no other message being sent on x, i.e. there is only one sender, and there is at least one receiver).
3. Subsequently, it waits to receive a message $r\,t\,f$ on the channel s. As we will see below, many x-receivers may send such a message. Intuitively, the channel r is the "back channel" to the x-receiver, the channel t stands for the truth-value "true", and the channel f stands for the truth value "false", in the context of the channel r.
4. The sender acknowledges the very first message it receives positively ; it sends the channel t on r, and sends the channel y on t.
5. In parallel, the sender sets up a persistent rule which responds with "false" to all messages (if any) from x-receivers which involve a different r channel than the one on which it has already sent a "true" acknowledgment.

Note that the sender is *not* modeled as a recursive process : a sender has finitely many sub-agents (some of which are persistent). An important property of the (translation of) the sender is that it will respond to any number of messages on the channel s it has broadcast on x. However, only the first will be acknowledged positively ; the others will all be acknowledged negatively.

A receiver behaves in a dual fashion :

1. It waits for a message s on the channel x.
2. It *selects one*, and proposes to commit to this sender, by sending it three new channels r, t and f. Note that the only receiver on s is a single ($x-$)sender. Many $x-$receivers may receive the same message s on x, and will respond by sending their triple. As we have discussed above, the x-sender will respond positively to only one of these triples.
3. It waits for an acknowledgment z on r from the chosen x-sender.
4. If the acknowledgment is positive, it waits for y on z, and then reduces to P.
5. If the acknowledgment is negative, then it is not participating in the current reduction, and it recursively goes back to wait for another message on x.

Note that in the recursive call, the receiver may propose once again to the same sender s, even though s has already reduced. Hence the encoding introduces the possibility of tau-loops (divergences).