Bisimulation dans CCS (2023)

Sujet
*

Le langage CCS (*Calculus of Communication Systems*) est un langage de description de processsus concurrents. On s'intéresse d'abord à une version simplifiée du langage, dont la syntaxe, construire à partir d'un ensemble infini d'*actions* a, b, c ... est donnée par :

$$\begin{array}{lll} P, Q & ::= & 0 \mid P \,||\, Q \mid \alpha.P \mid P + Q \\ \alpha & ::= & a \mid \overline{a} \end{array}$$

0 est le processus inactif; $P \,||\, Q$ est la composition parallèle des processus P et Q; $a.P$ (respectivement $\overline{a}.P$) est le processus qui effectue l'action a (respectivement le dual de l'action a) puis se comporte comme P; $P + Q$ est le choix non-déterministe entre P et Q.
Une relation de *congruence structurelle* $\equiv$ permet d'identifier des processus qui s'écrivent différemment mais représentent le même objet :

- $P \,||\, Q \equiv Q \,||\, P$ et $P \,||\, (Q \,||\, R) \equiv (P \,||\, Q) \,||\, R$ et $0 \,||\, P \equiv P$;
- $P + Q \equiv Q + P$ et $P + (Q + R) \equiv (P + Q) + R$ et $0 + P \equiv P$;
- si $P \equiv P'$, alors $\alpha.P \equiv \alpha.P'$;
- si $P \equiv P'$ et $Q \equiv Q'$, alors $P \,||\, Q \equiv P' \,||\, Q'$ et $P + Q \equiv P' + Q'$;
- $\equiv$ est réflexive, symétrique et transitive.

Dans la suite, on considérera les processus « modulo $\equiv$ », c'est-à-dire qu'on manipulera des classes d'équivalence de processus pour $\equiv$.
Une *étiquette* ℓ est soit une action (ou une action duale) α soit une *synchronisation* τ. La sémantique de CCS (la manière dont les processus évoluent) est donnée par la relation de *transition étiquetée* suivante :

- $\alpha.P \to^{\alpha} P$ (l'action α est jouée);
- $a.P \,||\, \overline{a}.Q \to^{\tau} P \,||\, Q$ (l'action a et l'action duale $\overline{a}$ se synchronisent);
- si $P \to^{\ell} P'$, alors $P \,||\, Q \to^{\ell} P' \,||\, Q$;
- si $P \to^{\ell} P'$, alors $P + Q \to^{\ell} P'$;
- si $P \to^{\ell} P'$ et $P \equiv Q$ et $P' \equiv Q'$, alors $Q \to^{\ell} Q'$.

On omettra les occurrences de 0 en fin de processus, par exemple on écrira $a.b$ plutôt que $a.b.0$. On considérera que le préfixe . est prioritaire sur la somme $+$ qui est prioritaire sur la composition parallèle $||$: ainsi, $a.b + c \,||\, d$ est $((a.b) + c) \,||\, d$.

Un *graphe de transition de* P est une représentation des processus accessibles depuis P par transitions successives sous forme de graphe où :

- les sommets sont les classes d'équivalence pour $\equiv$ de processus différents;
- les arêtes des transitions étiquetées entre les processus.

À titre d'exemple, le graphe de transition du processus $a.\overline{b} \,||\, \overline{a}$ est donnée dans la figure 1. On constate, entre autres, que $a.\overline{b} \,||\, \overline{a}$ peut effectuer trois transitions :

- $a.\overline{b} \,||\, \overline{a} \to^{a} \overline{b} \,||\, \overline{a}$ (on a joué l'action a);
- $a.\overline{b} \,||\, \overline{a} \to^{\overline{a}} a.\overline{b} \,||\, 0$ (on a joué l'action duale $\overline{a}$);
- $a.\overline{b} \,||\, \overline{a} \to^{\tau} \overline{b} \,||\, 0$ (on a synchronisé a et $\overline{a}$).

Question 1

Donner les graphes de transition des cinq processus suivants :

$$(1)\quad 0 \qquad (2)\quad a.b + a.c \qquad (3)\quad a.(b + c) \qquad (4)\quad a.\overline{a} + \overline{a}.a \qquad (5)\quad a \,||\, \overline{a}$$

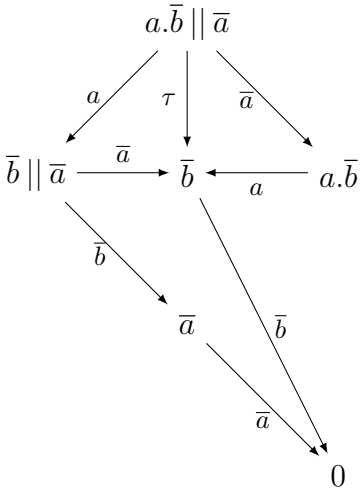

Figure 1.

Question 2

Montrer qu'un graphe de transition d'un processus est fini.

On ajoute la récursion à la syntaxe du langage : $P, Q \ ::= \ [...] \mid X \mid \mu X.P$ avec X appartenant à un ensemble infini de *variables de récursion*, avec la règle de congruence structurelle suivante :

$$\mu X.P \equiv P[\mu X.P/X]$$

où $P[Q/X]$ est le processus obtenu en remplaçant dans P chaque occurrence de X par le processus Q. En outre, si $P \equiv P'$, alors $\mu X.P \equiv \mu X.P'$.

Par exemple, on a $\mu X.a.X \equiv a.\mu X.a.X \equiv a.a.\mu X.a.X \equiv \ldots$

Question 3

Donner le graphe de transition des processus suivants :

$$(1) \quad \mu X.(a.X + b) \qquad (2) \quad a.\mu X.b.a.X \qquad (3) \quad \mu Y.a.b.a.b.Y$$

Question 4

Montrer qu'un graphe de transition peut être infini.

Un ensemble E et une relation d'ordre $\leqslant$ sur E forment un *treillis complet* $(E, \leqslant)$ quand toute partie $S \subseteq E$ admet une borne inférieure et une borne supérieure pour $\leqslant$.

Soit f croissante sur un treillis complet $(E, \leqslant)$.

Question 5

Montrer que si x est un point pré-fixe de f, c'est-à-dire si $x \leqslant f(x)$, alors $f(x)$ en est aussi un.

Question 6

Montrer que la borne supérieure des points pré-fixes de f est le plus grand point fixe de f.

Question 7

Montrer le lemme de Knaster-Tarski :

> Toute fonction f croissante sur un treillis complet admet un plus petit et un plus grand point fixe.

Soit $\mathcal{P}$ l'ensemble des processus de CCS et $\mathcal{L}$ l'ensemble des étiquettes. Une relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ est une *bisimulation* si pour tout couple $(P, Q) \in \mathcal{R}$:

- pour tout $(\ell, P') \in \mathcal{L} \times \mathcal{P}$, si $P \to^{\ell} P'$, alors il existe $Q' \in \mathcal{P}$ tel que $Q \to^{\ell} Q'$ et $(P', Q') \in \mathcal{R}$.
- pour tout $(\ell, Q') \in \mathcal{L} \times \mathcal{P}$, si $Q \to^{\ell} Q'$, alors il existe $P' \in \mathcal{P}$ tel que $P \to^{\ell} P'$ et $(P', Q') \in \mathcal{R}$.

La bisimilarité (notée $\sim$) est la plus grande bisimulation (pour l'inclusion).

La définition de la bisimilarité n'est pas inductive : elle est récursive, mais ne contient pas de cas de base (on dit qu'elle est *coinductive*).

Question 8

Montrer que la bisimilarité peut être formellement définie, de manière unique, par le théorème de Knaster-Tarski.

Question 9

Montrer que les processsus (2) et (3) de la question 3 sont *bisimilaires*, c'est-à-dire qu'ils appartiennent (en tant que couple de processus) à la bisimilarité.

Question 10

Montrer que les processus (2) et (3) de la question 1 ne sont pas bisimilaires.