

Partiel: Paradigmes de Programmation Concurrente 2016

Partie π -calcul

C.Agon, R.Demangeon

Demi-part à traiter sur une copie séparée. Tous documents papier autorisés. Barème à titre indicatif.

Exercice 1 – Réduction (1.5pts)

Donner les schémas de **réductions**¹ des processus suivants:

$$\begin{aligned} S_1 &= a(x).a(y).b(z).\bar{x}(z) \mid \bar{a}(u) \mid \bar{a}(v) \mid \bar{b}(u) \\ S_2 &= a(x).!x.\bar{x} \mid \bar{b} \mid \bar{a}(b) \mid \bar{a}(c) \\ S_3 &= !a(x).\bar{b}(x) \mid \bar{b}(a) \mid c(y,z).!y(t).\bar{z}(t) \mid \bar{c}(b,a) \end{aligned}$$

Exercice 2 – Modelisation (1pt)

Donner un processus π -calcul représentant un serveur qui a le comportement suivant:

- le serveur, toujours disponible, reçoit des requêtes sur une adresse fixe,
- les requêtes contiennent deux canaux d'interactions a et b et un canal de retour,
- après réception d'une requête, le serveur écoute simultanément sur les deux canaux a et b , correspondant à deux comportements **mutuellement exclusifs**:
- si il reçoit un entier sur le canal a , le serveur envoie sur le canal de retour le résultat du calcul d'une fonction f sur cet entier.
- si il reçoit deux entiers sur le canal b , il envoie le premier à un autre serveur (on ne demande pas de le représenter), récupère une réponse entière et applique une fonction g aux deux arguments suivants: (i) la réponse qu'il vient de récupérer et (ii) le deuxième entier passé sur b ; enfin, il envoie le résultat de ce calcul sur le canal de retour.

Exercice 3 – Bisimulation (2.5pts)

Donner l'équivalence² la plus forte entre les processus CCS suivants (justifier):

- $P_1 = a \mid \bar{a}$ et $Q_1 = a.\bar{a} + \bar{a}.a$
- $P_2 = P(a,b)$ avec $P(x,y) = x.P(y,x) + y.P(y,x)$ et $Q_2 = Q(a) \mid Q(b)$ avec $Q(z) = z.(z \mid Q(z))$
- $P_3 = a + a.b$ et $Q_3 = (\nu c) (!c.\bar{c} \mid a.b \mid a.\bar{c})$

¹un schéma contenant les différents processus obtenus en réduisant le processus initial, reliés par des flèches indiquant la réduction.

²équivalence de traces, bisimilarité faible ou bisimilarité forte

Exercice 4 – Types Simples (1pt)

Donner, quand c'est possible, des types simples aux canaux des processus S_1 , S_2 et S_3 .

Exercice 5 – Types pour la terminaison (2pts)

En suivant le principe du système de types de Deng-Sangiorgi pour la terminaison, examiner les deux processus suivants et:

- donner des types avec niveaux aux canaux du processus qui termine,
- expliquer pourquoi il est impossible de donner de tels types pour le processus qui diverge.

$$\begin{aligned} R_1 &= c(y).!y.\bar{b} \mid c(z).!z.\bar{a} \mid \langle a \rangle \mid \bar{c}\langle a \rangle \mid \bar{c}\langle b \rangle \\ R_2 &= !a(x).\bar{x} \mid !b.\bar{a}\langle v \rangle \mid d(z).\bar{z} \mid \bar{d}\langle b \rangle \end{aligned}$$

Exercice 6 – Sessions bien-formées (2pts)

Soit la session suivante:

$$A \rightarrow B : \text{hello}. B \rightarrow A : \begin{cases} \text{matin. } C \rightarrow A : \text{Bonjour. end} \\ \text{soir. } C \rightarrow A : \text{Bonsoir. end} \end{cases}$$

Question 1

Donner la projection de la session du point de vue de B, puis du point de vue de C.

Question 2

Expliquer pourquoi cette session est mal-formée, c'est à dire pourquoi elle ne modélise pas un comportement réaliste (entre programmés distants s'échangeant des messages au sein d'un réseau).

Question 3

Modifier la session pour qu'elle décrive un protocole similaire, mais qui corresponde à un comportement réaliste.

Question 4

Proposer les grandes lignes d'une analyse statique qui permet de détecter les sessions mal-formées.