Concurrency (partial exam 2009-2010, first part)

You may consult the slides and handouts of the lectures. No other document or electronic device is allowed. Answers should be formulated in French or English, and preferably in a rigorous and sharp style. Write the solutions on a sheet different from the one used for the second part of the exam. Your corrected copy can be consulted in Amadio's office, Chevaleret, 6C12.

Recall that $\mathcal{L}$ denotes the set of CCS names $a, b, \ldots$ and co-names $\bar{a}, \bar{b}, \ldots$ while *Act* denotes the set of CCS *actions* (i.e., $\mathcal{L} \cup \{\tau\}$). Let $s = \alpha_1\alpha_2\ldots\alpha_n \in Act^*$. Define $\stackrel{s}{\Longrightarrow}$ as $(\stackrel{\tau}{\longrightarrow})^* \stackrel{\alpha_1}{\longrightarrow} (\stackrel{\tau}{\longrightarrow})^* \ldots (\stackrel{\tau}{\longrightarrow})^* \stackrel{\alpha_n}{\longrightarrow} (\stackrel{\tau}{\longrightarrow})^*$. For the empty sequence $s = \epsilon$, $\stackrel{s}{\Longrightarrow}$ is defined as $(\stackrel{\tau}{\longrightarrow})^*$.

For the sake of simplicity in the first part of this exam we will restrict our attention to the following *finite (or recursion-free) CCS* fragment. Processes in this fragment are given by the following syntax:

$$P, Q := \sum_{i \in I} \alpha_i.P_i \quad | \quad P \mid Q \quad | \quad (\nu a)P.$$

where I is a finite indexing set and each $\alpha_i \in \mathcal{L}$. So, we only have summation, parallel composition, and restriction. (Recall that 0 denotes an empty summation, i.e., a summation where $|I| = 0$. Analogously, $\alpha.P$ is a summation where $|I| = 1$.)

In the following exercises we will recall the notions of trace equivalence and language equivalences studied in the second and fourth lectures but *restricted to finite CCS.*

Trace Equivalence

Definition 1 *The (set of) traces of P is given by $T(P) = \{s \in \mathcal{L}^* \mid P \stackrel{s}{\Longrightarrow}\}$. We say that P and Q are trace equivalent, written $P \sim_t Q$, iff $T(P) = T(Q)$.*

In the following exercises you will be asked to provide a compositional definition of $T(P)$. I.e. given P, you will be asked to determine the traces of P solely from the traces of the sub-terms of P.

Exercise 1 One can determine the traces of $\sum_{i \in I} \alpha_i.P_i$ from the sets $T(P_i)$ for each $i \in I$. In fact, show that the following equation

$$T(\sum_{i \in I} \alpha_i.P_i) = \{\epsilon\} \cup \bigcup_{i \in I} \{\alpha_i s \in \mathcal{L}^* \mid s \in T(P_i)\}$$

is valid, i.e., it holds. (Assume that $\bigcup_{i \in \emptyset} S = \emptyset$ for any set S.)

Exercise 2 Similarly, give a valid equation of the form

$$T((\nu a)P) = \ldots$$

to determine the traces of $(\nu a)P$ from the set $T(P)$.

Exercise 3 Give a valid equation of the form

$$T(P \mid Q) = \dots$$

to determine the traces of $P \mid Q$ from the sets $T(P)$ and $T(Q)$.

The above equations may actually help you prove that trace equivalence is a congruence.

Exercise 4 Show that if $P \sim_t Q$ then

1. $(\nu a)P \sim_t (\nu a)Q$,
2. $M + \alpha.P \sim_t M + \alpha.Q$ where M is a summation process,
3. $P \mid R \sim_t Q \mid R$.

Recall that trace equivalence allows you to equate processes with different deadlocked states: E.g., $a.0 + a.b.0 \sim_t a.b.0$ The notion of *language equivalence*, also known as *completed trace equivalence*, is more sensitive to this issue.

Language Equivalence (Completed-Trace Equivalence)

Recall from our lectures that a process generates a sequence of non-silent actions s if it can perform the actions of s in a finite maximal sequence of transitions. More precisely.

Definition 2 *The process P* generates *a sequence $s \in \mathcal{L}^*$ if and only if there exists Q such that $P \stackrel{s}{\Longrightarrow} Q$ and $Q \stackrel{\alpha}{\not\longrightarrow}$ for every $\alpha \in Act$. If P generates s we say that s is a* completed trace *of P. Define the language of (or generated by) a process P, $L(P)$, as the set of all sequences P generates. We say that P and Q are* language equivalent *(or completed-trace equivalent), written $P \sim_L Q$ iff $L(P) = L(Q)$.*

Unlike for trace equivalence we have $a.0 + a.b.0 \not\sim_L a.b.0$. In the last lecture (Révision) we already showed that $\sim_L$ is not a congruence for general CCS. We did this by illustrating that $\sim_L$ is not preserved under restriction. More precisely, we defined $R(a) \stackrel{\text{def}}{=} a.R(a)$ and $Div \stackrel{\text{def}}{=} \tau.Div$ and showed that $R(a) \mid b.0 \sim_L Div$ but $(\nu a)(R(a) \mid b.0) \not\sim_L (\nu a)Div$. Notice, however that $R(a)$ and Div are not finite CCS processes. We may therefore wonder if $\sim_L$ is also not preserved under restriction in the context of finite CCS.

Exercise 5. Does $(\nu a)P \sim_L (\nu a)Q$ hold for all finite CCS processes P and Q such that $P \sim_L Q$? If it does then prove it and if it doesn't, give a counter-example.

Solutions

Exercise 1. Let $P = \sum_{i \in I} \alpha_i.P_i$. Suppose $I = \emptyset$.. Then clearly $T(0) = \{s \mid 0 \overset{s}{\Longrightarrow}\} = \{\epsilon\} = \{\epsilon\} \cup \emptyset$. Now suppose $I \neq \emptyset$. With the help of rule SUM one can verify that whenever $P \overset{\alpha s'}{\Longrightarrow}$ then there must exists $j \in I$ so that $\alpha = \alpha_j$ and $P_j \overset{s'}{\Longrightarrow}$. Thus we get $T(P) = \{s \mid P \overset{s}{\Longrightarrow}\} = \{\epsilon\} \cup \{\alpha.s' \mid P_j \overset{s'}{\Longrightarrow} \ \wedge \alpha = \alpha_j \text{ for some } j \in I\} = \{\epsilon\} \cup \bigcup_{i \in I}\{\alpha_i s \in \mathcal{L}^* \mid s \in T(P_i)\}$.

Exercise 2. Define $\alpha \in s$ iff α occurs in s. The following is valid equation for $T((\nu a)P)$ in terms of $T(P)$:

$$T((\nu a)P) = \{s \in \mathcal{L}^* \mid s \in T(P) \wedge a \notin s \wedge \bar{a} \notin s\}.$$

This can be verified with the help of rule REST and the fact that for any Q, $T(Q)$ is prefixed-closed.

Exercise 3 The following is valid equation for $T(P \mid Q)$ in terms of $T(P)$ and $T(Q)$:

$$T(P \mid Q) = \{s \mid s \in \mathit{shuffle}(t, r) \text{ for some } t, r \text{ with } t \in T(P) \wedge r \in T(Q)\}$$

where $\mathit{shuffle}(t, r)$ is defined thus:

$$\begin{array}{rcl} \mathit{shuffle}(\epsilon, r) & = & \{r\} \\ \mathit{shuffle}(t, \epsilon) & = & \{t\} \\ \mathit{shuffle}(\alpha.t', \beta.r') & = & \{\alpha.s' | s' \in \mathit{shuffle}(t', \beta.r')\} \\ & & \cup\{\beta.s' | s' \in \mathit{shuffle}(\alpha.t', r')\} \\ & & \cup\{s' | \alpha = \bar{\beta} \wedge s \in \mathit{shuffle}(t', r')\} \end{array}$$

(The first two sets in the union correspond to all the interleaving of traces, while the last one correspond to synchronization along the traces)

Exercise 4 It follows from compositional definition of $T(P)$ given in the exercises above. Suppose that $P \sim_t Q$. We only show (4.3). We have

$$T(P \mid R) = \{s \mid s \in \mathit{shuffle}(t, r) \text{ for some } t, r \text{ with } t \in T(P) \wedge r \in T(R)\}$$

Since $T(P) = T(Q)$,

$$T(P \mid R) = \{s \mid s \in \mathit{shuffle}(t, r) \text{ for some } t, r \text{ with } t \in T(Q) \wedge r \in T(R)\} = T(Q \mid R)$$

as wanted.

Exercise 5 Language equivalence is not preserved under restriction. Take $P = a.b.0 + a.c.0$ and $Q = a.(b.0 + c.0)$. Notice that $L(P) = L(Q)$. However $a \in L((\nu c)P)$ but $a \notin L((\nu c)Q)$.