Concurrency – Lecture 10

Exercises relative to Lectures 5–9

December 4, 2003

Exercise 1: Equational reasoning in CCS

Consider the following three CCS processes, where the channel names a, b, c, d, e, f are supposed to be all different:

$$
\begin{array}{rcl}
P & \stackrel{\text{def}}{=} & (\nu c)(a.b.c.0 \mid d.\bar{b}.e.0) \\
Q & \stackrel{\text{def}}{=} & (\nu f)(a.f.c.0 \mid d.\bar{f}.e.0) \\
R & \stackrel{\text{def}}{=} & (\nu c)(a.c.0 \mid d.e.0)
\end{array}
$$

Show that

$$Ax \vdash P = Q \quad \text{while} \quad Ax \not\vdash P = R$$

where Ax are the standard axioms of the theory of CCS (for observational equivalence).

Exercise 2: Specifications in CCS

Consider the following (infinite) specification of an unbounded queue in value-passing CCS, where ":" stands for the concatenation operation of an element to a sequence, and "ϵ" stands for the empty sequence. In this specification, *in*, *out* and *empty* represent channel names.

$$
\begin{array}{rcl}
Queue^{\epsilon} & \stackrel{\text{def}}{=} & in(x).Queue^{x} + empty.Queue^{\epsilon} \\
Queue^{q:v} & \stackrel{\text{def}}{=} & in(x).Queue^{x:q:v} + \overline{out}(v).Queue^{q}
\end{array}
$$

The exercise consists in defining a finite "implementation" for the queue, still in value-passing CCS, that corresponds (i.e. is weakly bisimilar to) the above specification. You don't need to prove weakly bisimilarity.

Hint: The basic idea is similar to the implementation of the Counter illustrated in Lectures 6-7.

Exercise 3: Early and late bisimulation in the π-calculus

Consider an extension of the π-calculus with the so-called match operator $[x = y]P$, whose semantics (early and late) is defined by

$$\frac{P \xrightarrow{\mu} Q}{[x = x]P \xrightarrow{\mu} Q}$$

(note that there are no transitions from $[x = y]P$ when $x \neq y$).

Consider the processes

$$\begin{aligned} P &\stackrel{\text{def}}{=} x(y).\bar{z}w.0 + x(y).0 \\ Q &\stackrel{\text{def}}{=} x(y).\bar{z}w.0 + x(y).0 + x(y).[y = z]\bar{z}w.0 \end{aligned}$$

1. Show that P and Q are early bisimilar but not late bisimilar.
2. Define a Q with similar properties without using the match operator

Exercise 4: Recursion vs iteration in the π-calculus

Consider a variant of the π-calculus, $R\pi$, where the iteration operator is replaced by recursion, i.e. there is no "!" and processes can contain process names (with parameters which represent channel names) $A(x)$, defined by equations like

$$A(x) \stackrel{\text{def}}{=} P$$

where P is a process in $R\pi$.

The semantics of recursion is defined in the usual way, namely:

$$\frac{P[y/x] \xrightarrow{\mu} Q}{A(y) \xrightarrow{\mu} Q}$$

in both early and late semantics

1. Define an encoding $[\![\cdot]\!] : R\pi \longrightarrow \pi$ which is fully abstract wrt late bisimulation. **Hint:** Use ! to expand a copy of P whenever needed. In order to avoid that P is activated also when not needed, use an input prefix on a fresh restricted channel. The call of P, namely $A(y)$, can then be simulated by an output action on the same channel, with y as parameter.
2. Prove the full-abstraction of the encoding wrt late bisimulation.