Concurrency problems class

Catuscia Palamidessi, Jean-Jacques Lévy, James J. Leifer

Catuscia@lix.polytechnique.fr, Jean-Jacques.Levy@inria.fr, James.Leifer@inria.fr

11 December 2003

1 Definition: CCS processes

$P ::=$	0	empty
	$\alpha.P$	prefixing
	$P \mid P$	parallel composition
	$(\nu\, L)P$	hiding
	$P + P$	summation
....	....	....
	K	constant (for expressing recursion)
	$!P$	π-calculus-style replication (for expressing recursion)
	$\mu X.P$	fixed-point (for expressing recursion)

2 Definition: CCS alphabetic conventions

a	name
$\overline{a}$	co-name
ℓ	label (ranges over names and co-names)
L	label set
f	label map
α	action (ranges over labels and τ)

3 Definition: CCS labelled transitions rules

- input: $a.P \xrightarrow{a} P$
- output: $\overline{a}.P \xrightarrow{\overline{a}} P$
- synchronization: $\dfrac{P \xrightarrow{\ell} P' \qquad Q \xrightarrow{\overline{\ell}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$
- choice: $\dfrac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$ (and symmetrically)
- parallel composition: $\dfrac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$ (and symmetrically)
- hiding: $\dfrac{P \xrightarrow{\alpha} P'}{(\nu\, L)P \xrightarrow{\alpha} (\nu\, L)P'}$ if $\alpha, \overline{\alpha} \notin L$
- and others, for example...
- constant: $\dfrac{P \xrightarrow{\alpha} P'}{K \xrightarrow{\alpha} P'}$ if $K = P$
- replication (many possible): $\dfrac{P|!P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'}$

- fixed-point (many possible): $\dfrac{P\{\mu X.P/X\} \xrightarrow{\alpha} P'}{\mu X.P \xrightarrow{\alpha} P'}$

4 Definition: CCS operational equivalences

- strong simulation: a relation $\mathcal{R}$ is a strong simulation if for all $(P, Q) \in \mathcal{R}$ and $P \xrightarrow{\alpha} P'$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$.
- strong bisimulation: a relation $\mathcal{R}$ is a strong bisimulation if it and its inverse are strong simulations.
- strong bisimilarity: $\sim$ is the largest strong bisimulation.
- weak simulation: a relation $\mathcal{R}$ is a weak simulation if for all $(P, Q) \in \mathcal{R}$ we have:
  1. if $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \xrightarrow{\tau}{}^{*} Q'$ and $(P', Q') \in \mathcal{R}$.
  2. if $P \xrightarrow{\ell} P'$ then there exists Q' such that $Q \xrightarrow{\tau}{}^{*} \xrightarrow{\ell} \xrightarrow{\tau}{}^{*} Q'$ and $(P', Q') \in \mathcal{R}$.
- weak bisimulation: a relation $\mathcal{R}$ is a weak bisimulation if it and its inverse are weak simulations.
- weak bisimilarity (also known as bisimilarity, also known as observational equivalence): $\approx$ is the largest weak bisimulation.
- observational congruence: $\cong$ is the largest symmetric relation satisfying the following property: if $P \cong Q$ and $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \xrightarrow{\tau}{}^{*} \xrightarrow{\ell} \xrightarrow{\alpha}{}^{*} Q'$ and $P' \approx Q'$.

5 Exercise (CCS): unreliable transmission medium

A transmitter T, an unrealiable transmission medium M, and a receiver R are modelled as follows:

$$\begin{array}{lcl} T & \stackrel{\text{def}}{=} & in.\bar{i}.T' \\ T' & \stackrel{\text{def}}{=} & r.\bar{i}.T' + a.T \\ M & \stackrel{\text{def}}{=} & i.M' \\ M' & \stackrel{\text{def}}{=} & \overline{o}.M + \tau.\overline{r}.M \\ R & \stackrel{\text{def}}{=} & o.\overline{out}.\overline{a}.R \end{array}$$

M is an unreliable medium: having received an input message from T (action i) it either outputs the message to R (action $\overline{o}$), or loses it (action τ) and then sends a request for retransmission (action $\overline{r}$). If R does receive the message, it delivers it (action $\overline{out}$) and sends an acknowledgement directly to T (action $\overline{a}$).

1. Calculate the transition graph of $(\nu\, i, o, r, a)(T|M|R)$ and hence show that this process is observationally equivalent to a simple reliable buffer B defined by:

$$B \stackrel{\text{def}}{=} in.\overline{out}.B$$

2. Are $(\nu\, i, o, r, a)(T|M|R)$ and B observationally congruent?
3. Do the two have the same behavior with respect to *divergence*, that is can either perform a series of actions ending in an infinite sequence of τ actions?

6 Exercise (CCS): semaphores

1. A *semaphore* is a mechanism to prevent more than a certain number n of clients from simultaneously entering their *critical sections* to access a precious resource. A client "brackets" its critical section by requesting entry permission (action $\overline{wait}$) and then signaling when it is finished (action $\overline{signal}$):

   $$\overline{wait} \quad \text{...critical section...} \quad \overline{signal}$$

   Note that a mutual exclusion lock (mutex) is a special case (when $n = 1$) of a semaphore.

   Define a CCS process to model a semaphore of capacity n. Hint: create a constant Sem^n_k, for $0 < n$ and $0 \leq k \leq n$, that represent a semaphore in the state when k clients are in their critical sections. You will need to treat the cases $k = 0$ and $k = n$ specially.

7 Exercise (CCS): deadlock

We say that a process *can deadlock* if it can perform a sequence of actions to enter a state that is observationally congruent ($\cong$) to 0.

Let

$$\begin{aligned} C &\stackrel{\text{def}}{=} g_0.g_1.p_0.p_1.C \\ D &\stackrel{\text{def}}{=} g_1.g_0.p_1.p_0.D \\ S_0 &\stackrel{\text{def}}{=} \overline{g_0}.\overline{p_0}.S_0 \\ S_1 &\stackrel{\text{def}}{=} \overline{g_1}.\overline{p_1}.S_1 \end{aligned}$$

1. For each of the following processes, determine whether or not it can deadlock:

   $$(\nu\, g_0, p_0, g_1, p_1)(C|C|S_0|S_1)$$
   $$(\nu\, g_0, p_0, g_1, p_1)(C|D|S_0|S_1)$$

2. Prove that $P \cong 0$ iff P can do no action.
3. Prove that $T \approx 0$ where $T \stackrel{\text{def}}{=} \tau.T$.
4. Hence show that it is possible for a process that can deadlock to be observationally congruent to one that cannot deadlock.

8 Exercise (π-calculus): arithmetic

We can define a process N_n for representing the natural number n as follows:

$$\begin{aligned} N_0(s, z) &\stackrel{\text{def}}{=} \overline{z} \\ N_{n+1}(s, z) &\stackrel{\text{def}}{=} \overline{s}.N_n(s, z) \end{aligned}$$

Thus $N_n(s, z)$ ouputs n times on s and then outputs on z.

Our goal is define a process $A(s_0, z_0, s_1, z_1, s, z)$ for adding numbers which has the property that

$$(\nu\, s_0, z_0, s_1, z_1)(N_{n_0}(s_0, z_0)|N_{n_1}(s_1, z_1)|A(s_0, z_0, s_1, z_1, s, z)) \approx N_{n_0+n_1}(s, z) \qquad (*)$$

- First define a processes $C(s, z, s', z')$ for copying a number from (s, z) to (s', z') and prove that

  $$(\nu\, s, z)(N_n(s, z)|C(s, z, s', z')) \approx N_n(s', z')$$

- Then define addition $A(s_0, z_0, s_1, z_1, s, z)$ and prove $(*)$ above.