Concurrency 5

CCS - Up-to bisimulation. Weak bisimulation. Axiomatizations.

Catuscia Palamidessi
INRIA Futurs and LIX - Ecole Polytechnique

The other lecturers for this course:

Jean-Jacques Lévy (INRIA Rocquencourt)
James Leifer (INRIA Rocquencourt)
Eric Goubault (CEA)

http://pauillac.inria.fr/~leifer/teaching/mpri-concurrency-2005/

Outline

A brief summary: Syntax of CCS

- (channel, port) names: $a, b, c, \ldots$
- co-names: $\overline{a}, \overline{b}, \overline{c}, \ldots$ Note: $\overline{\overline{a}} = a$
- silent action: τ
- actions, prefixes: $\mu ::= a \mid \overline{a} \mid \tau$
- processes: P, Q ::=

::=	0	inaction
|	$\mu.P$	prefix
|	$P \mid Q$	parallel
|	$P + Q$	(external) choice
|	$(\nu a)P$	restriction
|	$K(\vec{a})$	process name with parameters

- Process definitions: $D ::= K(\vec{x}) \stackrel{\mathrm{def}}{=} P$ where $fn(P) \subseteq \vec{x}$
- $fn(P)$ is the set of free (channel) names in P (occurrences not in the scope of ν)
- conversely, $bn(P)$ is the set of bound names in P (occurrences in the scope of ν)

A brief summary: Labeled transition system for CCS

We assume a given set of definitions D

$$[\text{Act}]\ \frac{}{\mu.P \xrightarrow{\mu} P} \qquad [\text{Res}]\ \frac{P \xrightarrow{\mu} P' \quad \mu \neq a, \overline{a}}{(\nu a)P \xrightarrow{\mu} (\nu a)P'}$$

$$[\text{Sum1}]\ \frac{P \xrightarrow{\mu} P'}{P+Q \xrightarrow{\mu} P'} \qquad [\text{Sum2}]\ \frac{Q \xrightarrow{\mu} Q'}{P+Q \xrightarrow{\mu} Q'}$$

$$[\text{Par1}]\ \frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \qquad [\text{Par2}]\ \frac{Q \xrightarrow{\mu} Q'}{P|Q \xrightarrow{\mu} P|Q'}$$

$$[\text{Com}]\ \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad [\text{Rec}]\ \frac{P[\vec{a}/\vec{x}] \xrightarrow{\mu} P' \quad K(\vec{x}) \stackrel{\mathrm{def}}{=} P \in D}{K(\vec{a}) \xrightarrow{\mu} P'}$$

A brief summary: (Strong) Bisimulation and Bisimilarity

- **Definition** We say that a relation $\mathcal{R}$ on processes is a *bisimulation* if

$$P \mathrel{\mathcal{R}} Q \quad \text{implies that} \quad \begin{array}{l} \textit{if } P \xrightarrow{\mu} P' \textit{ then } \exists Q' \textit{ s.t. } Q \xrightarrow{\mu} Q' \textit{ and } P' \mathrel{\mathcal{R}} Q' \\ \textit{if } Q \xrightarrow{\mu} Q' \textit{ then } \exists P' \textit{ s.t. } P \xrightarrow{\mu} P' \textit{ and } P' \mathrel{\mathcal{R}} Q' \end{array}$$

- Note that this property does not uniquely defines $\mathcal{R}$. There may be several relations that satisfy it.
- **Definition** (Bisimilarity) $\sim \; = \; \bigcup\{\mathcal{R} \mid \mathcal{R} \text{ is a bisimulation}\}$
- **Theorem** $\sim$ is a bisimulation.
- $P \sim Q$ (P is bisimilar to Q) intuitively means that Q can do everything that P can do, and vice versa, at every step of the computation.

A brief summary: The coinductive method

- Bisimilarity is a coinductive definition.
- In order to prove that $P \sim Q$ it is sufficient to find a bisimulation $\mathcal{R}$ such that $P \mathrel{\mathcal{R}} Q$

Bisimilarity in CCS is a congruence

- **Definition** A relation $\mathcal{R}$ on a language is a *congruence* if
  - $\mathcal{R}$ is an equivalence relation (i.e. it is reflexive, symmetric, and transitive), and
  - $\mathcal{R}$ is preserved by all the operators of the language, namely if $P \mathrel{\mathcal{R}} Q$ then $op(P_1, \ldots, P, \ldots, P_n) \mathrel{\mathcal{R}} op(P_1, \ldots, Q, \ldots, P_n)$
- **Theorem** $\sim$ is a congruence relation
- This is an important property as it allows to prove equivalence in a modular way.

Proof of the congruence theorem

- Bisimilarity is an equivalence relation.
  - Reflexivity. Let $\mathcal{R} = \{(P, P) \mid P \text{ is a CCS process }\}$. Then $\mathcal{R}$ is a bisimulation (Immediate).
  - Symmetry. If $P \mathrel{\mathcal{R}} Q$, with $\mathcal{R}$ bisimulation, then $Q \mathrel{\mathcal{R}^{-1}} P$ holds and $\mathcal{R}^{-1}$ is a bisimulation (Immediate).
  - Transitivity. If $P \mathrel{\mathcal{R}} Q$ and $Q \mathrel{\mathcal{S}} R$, with $\mathcal{R}, \mathcal{S}$ bisimulations, then $P \mathrel{\mathcal{R} \circ \mathcal{S}} R$ holds and $\mathcal{R} \circ \mathcal{S}$ is a bisimulation (Proof: exercise).
- We have to prove now that $\sim$ is preserved by the operators of CCS. We prove it for the most complicated case, the parallel operator, and we leave the others as exercise.

Proof of the congruence theorem: case of the parallel operator

Assume $P \sim Q$. We prove that for any R, $P \mid R \sim Q \mid R$ holds. (The case $R \mid P \sim R \mid Q$ is analogous). Let

$$\mathcal{R} = \{(P' \mid R', Q' \mid R') \mid P', Q', R' \text{ are CCS processes and } P' \sim Q' \}$$

We only need to prove that $\mathcal{R}$ is a bisimulation.

- (Decomposition phase) Assume $P' \mid R' \xrightarrow{\mu} P'' \mid R''$. There are three cases, corresponding to the three rules for parallel composition.
  - (Rule Par1) In this case $P' \xrightarrow{\mu} P''$ and $R'' = R'$. Since $P' \sim Q'$, there exists Q'' such that $Q' \xrightarrow{\mu} Q''$ and $P'' \sim Q''$.
  - (Rule Par2) In this case $P'' = P'$ and $R' \xrightarrow{\mu} R''$. Take $Q'' = Q'$.
  - (Rule Com) In this case $P' \xrightarrow{a} P''$, $R' \xrightarrow{\bar{a}} R''$, and $\mu = \tau$. Since $P' \sim Q'$, there exists Q'' such that $Q' \xrightarrow{a} Q''$ and $P'' \sim Q''$.
- (Composition phase) In each of the three cases, $Q' \mid R' \xrightarrow{\mu} Q'' \mid R''$ for some Q'' such that $P'' \sim Q''$, and therefore $P'' \mid R'' \mathrel{\mathcal{R}} Q'' \mid R''$ holds.

Some interesting bisimilarities

The following bisimilarities hold

$$\begin{array}{rcl} P+0 & \sim & P \\ P+Q & \sim & Q+P \\ (P+Q)+R & \sim & P+(Q+R) \\ P+P & \sim & P \end{array} \qquad \begin{array}{rcl} P \mid 0 & \sim & P \\ P \mid Q & \sim & Q \mid P \\ (P \mid Q) \mid R & \sim & Q \mid (P \mid R) \end{array}$$

$$\begin{array}{rcll} (\nu a)0 & \sim & 0 & \\ (\nu a)(P \mid Q) & \sim & P \mid (\nu a)Q & \text{if } a \notin fn(P) \\ (\nu a)(P + Q) & \sim & P + (\nu a)Q & \text{if } a \notin fn(P) \\ (\nu a)(\nu b)P & \sim & (\nu b)(\nu a)P & \\ (\nu a)(P) & \sim & (\nu b)P[b/a] & \text{if } a \text{ not in the scope of } (\nu b) \text{ in } P\ (\alpha \text{ conversion}) \\ (\nu a)b.P & \sim & 0 & \text{if } b = a \text{ or } b = \bar{a} \\ (\nu a)b.P & \sim & b.(\nu a)P & \text{if } b \neq a \text{ and } b \neq \bar{a} \\ & & & \\ K(\vec{a}) & \sim & P[\vec{a}/\vec{x}] & \text{if } K(\vec{x}) \stackrel{\text{def}}{=} P \end{array}$$

Proof: Exercise

Some interesting bisimilarities: The Expansion Laws

The following bisimilarities hold

$$\begin{array}{rcl} a.P \mid b.Q & \sim & a.(P \mid b.Q) \ + \ b.(a.P \mid Q) \ \text{ if } b \neq \bar{a} \\ a.P \mid \bar{a}.Q & \sim & a.(P \mid \bar{a}.Q) \ + \ \bar{a}.(a.P \mid Q) \ + \ \tau.(P \mid Q) \end{array}$$

More in general:

$$(P \mid Q) \quad \sim \quad (\mu.\textstyle\sum_{P \xrightarrow{\mu} P'} P' \mid Q) \ + \ (\mu.\textstyle\sum_{Q \xrightarrow{\mu} Q'} P \mid Q') \ + \ (\tau.\textstyle\sum_{\substack{P \xrightarrow{a} P' \\ Q \xrightarrow{\bar{a}} Q'}} P' \mid Q')$$

Proof: Exercise

Axiomatization of strong bisimilarity

An equational theory for CCS: the axioms are the "=" version of the bisimilarities seen in previous pages. That is:

$$\begin{array}{rcl} P+0 & = & P \\ P+Q & = & Q+P \\ (P+Q)+R & = & P+(Q+R) \\ P+P & = & P \end{array} \qquad \begin{array}{rcl} P \mid 0 & = & P \\ P \mid Q & = & Q \mid P \\ (P \mid Q) \mid R & = & Q \mid (P \mid R) \end{array}$$

$$\begin{array}{rcll} (\nu a)0 & = & 0 & \\ (\nu a)(P \mid Q) & = & P \mid (\nu a)Q & \text{if } a \notin fn(P) \\ (\nu a)(P + Q) & = & P + (\nu a)Q & \text{if } a \notin fn(P) \\ (\nu a)(\nu b)P & = & (\nu b)(\nu a)P & \\ (\nu a)(P) & = & (\nu b)P[b/a] & \text{if } a \text{ not in the scope of } (\nu b) \text{ in } P\ (\alpha \text{ conversion}) \\ (\nu a)b.P & = & 0 & \text{if } b = a \text{ or } b = \bar{a} \\ (\nu a)b.P & = & b.(\nu a)P & \text{if } b \neq a \text{ and } b \neq \bar{a} \\ & & & \\ K(\vec{a}) & = & P[\vec{a}/\vec{x}] & \text{if } K(\vec{x}) \stackrel{\text{def}}{=} P \end{array}$$

$$(P \mid Q) \quad = \quad (\mu.\textstyle\sum_{P \xrightarrow{\mu} P'} P' \mid Q) \ + \ (\mu.\textstyle\sum_{Q \xrightarrow{\mu} Q'} P \mid Q') \ + \ (\tau.\textstyle\sum_{\substack{P \xrightarrow{a} P' \\ Q \xrightarrow{\bar{a}} Q'}} P' \mid Q')$$

Axiomatization of strong bisimilarity

To prove equivalence of recursive processes, it is convenient to introduce also the following conditional axiom (Unique solution of equations):

Given a context $C[\,]$ (that is, a term with some "holes" $[\,]$), in which all the holes are guarded (that is, occur in the context of a prefix), then, for every P and Q

$$\text{if} \quad P = C[P] \quad \text{and} \quad Q = C[Q] \quad \text{then} \quad P = Q$$

Definition We write $Ax \vdash P = Q$ iff $P = Q$ is derivable from the above axioms Ax and the usual rules for equality

Theorem (Soundness of the axiomatization) If $Ax \vdash P = Q$ then $P \sim Q$.

The converse of the above theorem does not hold, i.e. the axiomatization is not complete. Note that the existence of a sound and complete axiomatization would imply the decidability of bisimilarity (because the complement of the bisimarity relation is semidecidable). However, bisimilarity is not decidable in CCS.

The characterization of subsets of CCS in which bisimulation is decidable is an important research field in Concurrency Theory.

Bisimulation up-to $\sim$

The axiomatization can be combined with the coinductive method to ease proving bisimilarity. The idea is based on the so-called "bisimulation up-to $\sim$" technique.

- **Definition** We say that a relation $\mathcal{R}$ on processes is a *bisimulation up-to* $\sim$ if

$$P \,\mathcal{R}\, Q \quad \text{implies that} \quad \begin{array}{l} \textit{if } P \xrightarrow{\mu} P' \textit{ then } \exists Q' \textit{ s.t. } Q \xrightarrow{\mu} Q' \textit{ and } P' \sim \mathcal{R} \sim Q' \\ \textit{if } Q \xrightarrow{\mu} Q' \textit{ then } \exists P' \textit{ s.t. } P \xrightarrow{\mu} P' \textit{ and } P' \sim \mathcal{R} \sim Q' \end{array}$$

  Notation: $P' \sim \mathcal{R} \sim Q'$ means: $\exists P'', Q''$ s.t. $P' \sim P''$, $P'' \,\mathcal{R}\, Q''$, and $Q'' \sim Q'$.
  Note that in order to prove $P' \sim P''$ and $Q'' \sim Q'$, it is sufficient to prove $Ax \vdash P' = P''$ and $Ax \vdash Q'' = Q'$.
- **Theorem** If there exists $\mathcal{R}$ such that $\mathcal{R}$ is a bisimulation up-to $\sim$, and $P \,\mathcal{R}\, Q$ holds, then $P \sim Q$ holds.

The advantage of the above technique is that it is usually easier to define a relation that is a bisimulation up-to $\sim$, rather than a bisimulation.

Exercises

- Prove that $A \sim B$ where $A \stackrel{\text{def}}{=} a.A$ and $B \stackrel{\text{def}}{=} a.B + a.a.B$
- Prove that the two definitions of semaphores given in previous lecture are equivalent (that is, the semaphores given by those two definitions are bisimilar)

Value-passing CCS

In pure CCS processes communicate via channel (or port) names, but the communication does not carry data (or values). We now illustrate how we can modify CCS so that processes can send (and receive) values.

- (data) variables: $y, z, \ldots$
- (data) values: $v, w, \ldots$
- (channel, port) names: $a, b, c, \ldots$
- co-names: $\bar{a}, \bar{b}, \bar{c}, \ldots$
- silent action: τ
- actions, prefixes: $\mu ::= a\langle\vec{y}\rangle \mid \bar{a}\langle\vec{v}\rangle \mid \tau$
- processes:

$$\begin{array}{lcll} P, Q & ::= & 0 & \text{inaction} \\ & \mid & \mu.P & \text{prefix} \\ & \mid & P \mid Q & \text{parallel} \\ & \mid & P + Q & \text{(external) choice} \\ & \mid & (\nu a)P & \text{restriction} \\ & \mid & K(\vec{a})\langle\vec{v}\rangle & \text{process name with parameters} \end{array}$$

- Process definitions: $D \;::=\; K(\vec{x})\langle\vec{y}\rangle \stackrel{\text{def}}{=} P \quad \text{where } fn(P) \subseteq \vec{x}$

The labeled transition system for value-passing CCS

We assume a given set of definitions D

$$[\text{Send}]\ \frac{}{\overline{a}\langle\vec{v}\rangle.P \xrightarrow{a\langle\vec{v}\rangle} P}$$

$$[\text{Receive}]\ \frac{}{a\langle\vec{y}\rangle.P \xrightarrow{a\langle\vec{v}\rangle} P[\vec{v}/\vec{y}]} \qquad [\text{Res}]\ \frac{P \xrightarrow{\mu} P' \quad \mu{\neq}a,\overline{a}}{(\nu a)P \xrightarrow{\mu} (\nu a)P'}$$

$$[\text{Sum1}]\ \frac{P \xrightarrow{\mu} P'}{P{+}Q \xrightarrow{\mu} P'} \qquad [\text{Sum2}]\ \frac{Q \xrightarrow{\mu} Q'}{P{+}Q \xrightarrow{\mu} Q'}$$

$$[\text{Par1}]\ \frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \qquad [\text{Par2}]\ \frac{Q \xrightarrow{\mu} Q'}{P|Q \xrightarrow{\mu} P|Q'}$$

$$[\text{Com}]\ \frac{P \xrightarrow{a\langle\vec{v}\rangle} P' \quad Q \xrightarrow{\overline{a}\langle\vec{v}\rangle} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad [\text{Rec}]\ \frac{P[\vec{a}/\vec{x}][\vec{v}/\vec{y}] \xrightarrow{\mu} P' \quad K(\vec{x})\langle\vec{y}\rangle \stackrel{\text{def}}{=} P \in D}{K(\vec{a})\langle\vec{v}\rangle \xrightarrow{\mu} P'}$$

Bisimulation and bisimilarity are defined as usual, considering as labels those of the form $a\langle\vec{v}\rangle$ and $\overline{a}\langle\vec{v}\rangle$

Translating value-passing CCS into pure CCS

If the domain V of values is finite, we can emulate value-passing CCS by translating it into pure CCS in the following way. We denote this translation by

$$[\![\cdot]\!] : \text{ value-passing CCS } \rightarrow \text{ pure CCS}$$

For simplicity we assume channel arity 1.

$$\begin{array}{rcl}
[\![0]\!] & = & 0 \\
[\![\overline{a}\langle v\rangle.P]\!] & = & \overline{a_v}.[\![P]\!] \\
[\![a\langle y\rangle.P]\!] & = & \sum_{v\in V} a_v.[\![P[v/y]]\!] \\
[\![P \mid Q]\!] & = & [\![P]\!] \mid [\![Q]\!] \\
[\![P+Q]\!] & = & [\![P]\!] + [\![Q]\!] \\
[\![(\nu a)P]\!] & = & (\nu a)[\![P]\!] \\
[\![K(\vec{a})\langle v\rangle]\!] & = & K_v(\vec{a})
\end{array}$$

Furthermore, each definition $K(\vec{x})\langle\vec{y}\rangle \stackrel{\text{def}}{=} P$ is replaced by the following set of definitions, one for each $v \in V$:

$$K_v(\vec{x}) \stackrel{\text{def}}{=} [\![P[v/y]]\!]$$

Weak Bisimulation and Weak Bisimilarity

Motivation: abstract from internal actions (i.e. τ actions)

We introduce the following notation

- $P(\xrightarrow{\tau})^*Q$ iff $P = P_0 \xrightarrow{\tau} P_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} P_n = Q$
- $P \stackrel{\vec{\mu}}{\Rightarrow} Q$ iff
  $P = P_0\ (\xrightarrow{\tau})^* \xrightarrow{\mu_1} (\xrightarrow{\tau})^*\ P_1\ (\xrightarrow{\tau})^* \xrightarrow{\mu_2} (\xrightarrow{\tau})^* \ldots (\xrightarrow{\tau})^* \xrightarrow{\mu_n} (\xrightarrow{\tau})^*\ P_n = Q$
  and $\vec{\mu} = \mu_1\mu_2\ldots\mu_n$
- $P \stackrel{\widehat{\mu}}{\Rightarrow} Q$ iff
  $P \stackrel{\vec{\mu}}{\Rightarrow} Q$ and $\widehat{\mu}$ is obtained from $\vec{\mu}$ by removing all the τ's

Examples: $\widehat{a} = a$, $\widehat{\tau a \tau} = a$, $\widehat{\tau} = \varepsilon$ (empty string), $\widehat{a\tau b} = ab$.

Note:

- $P \stackrel{\widehat{a}}{\Rightarrow} Q$ means $P\ (\xrightarrow{\tau})^* \xrightarrow{a} (\xrightarrow{\tau})^*\ Q$,
- $P \stackrel{\widehat{\tau}}{\Rightarrow} Q$ means $P \stackrel{\varepsilon}{\Rightarrow} Q$, namely $P\ (\xrightarrow{\tau})^*\ Q$,
- $P \stackrel{\tau}{\Rightarrow} Q$ means $P\ (\xrightarrow{\tau})^* \xrightarrow{\tau} (\xrightarrow{\tau})^*\ Q$, namely $P\ (\xrightarrow{\tau})^+\ Q$ (at least one τ-step).

Weak Bisimulation and Weak Bisimilarity

- **Definition** We say that a relation R on processes is a *weak bisimulation* if

$$P\ \mathcal{R}\ Q \quad \text{implies that} \quad \begin{array}{l} \textit{if } P \xrightarrow{\mu} P' \textit{ then } \exists Q' \textit{ s.t. } Q \stackrel{\widehat{\mu}}{\Rightarrow} Q' \textit{ and } P'\ \mathcal{R}\ Q' \\ \textit{if } Q \xrightarrow{\mu} Q' \textit{ then } \exists P' \textit{ s.t. } P \stackrel{\widehat{\mu}}{\Rightarrow} P' \textit{ and } P'\ \mathcal{R}\ Q' \end{array}$$

- Note that this property does not uniquely defines $\mathcal{R}$. There may be several relations that satisfy it.
- **Definition** (Bisimilarity) $\approx\ =\ \bigcup\{\mathcal{R} \mid \mathcal{R}$ is a weak bisimulation$\}$
- **Theorem** $\approx$ is a weak bisimulation.
- $P \approx Q$ (P is weakly bisimilar to Q) intuitively means that at every step of the computation Q can do everything that P can do, and vice versa, if we ignore the internal actions.

Observation congruence

Unfortunately $\approx$ is not a congruence.
Example: $a.P \approx \tau.a.P$ but $a.P + b.Q \not\approx \tau.a.P + b.Q$

- **Definition** We say that P and Q are *observation-congruent* (notation $P \cong Q$) iff

$$\begin{array}{l} \textit{if } P \xrightarrow{\mu} P' \textit{ then } \exists Q' \textit{ s.t. } Q \stackrel{\mu}{\Rightarrow} Q' \textit{ and } P' \approx Q' \\ \textit{if } Q \xrightarrow{\mu} Q' \textit{ then } \exists P' \textit{ s.t. } P \stackrel{\mu}{\Rightarrow} P' \textit{ and } P' \approx Q' \end{array}$$

  The difference between $\approx$ and $\cong$ is when the first step is a τ-step. If $P \xrightarrow{\tau} P'$ and $P \approx Q$, then we can take $Q' = Q$ provided that $P' \approx Q'$. On the contrary, if $P \cong Q$, then we are obliged to find a Q' such that $Q \stackrel{\tau}{\Rightarrow} Q'$ (which means, at least one τ-step from Q to Q'), and $P' \approx Q'$.

- **Theorem** $\sim\ \subseteq\ \cong\ \subseteq\ \approx$

  Proof: exercise.

- **Theorem** $\cong$ is a congruence, and more precisely, it is the largest congruence on CCS contained in $\approx$.

Properties of observation congruence

Proposition The following properties hold:

$$\begin{array}{rcl} \mu.\tau.P & \cong & \mu.P \\ P + \tau.P & \cong & \tau.P \\ \mu.(P + \tau.Q) + \mu.Q & \cong & \mu.(P + \tau.Q) \end{array}$$

Proof: exercise.

The above properties are used to give an axiomatization observation congruence. The "=" version of the above are called τ-laws:

$$\begin{array}{rcl} \mu.\tau.P & = & \mu.P \\ P + \tau.P & = & \tau.P \\ \mu.(P + \tau.Q) + \mu.Q & = & \mu.(P + \tau.Q) \end{array}$$

Definition We write $Ax_\tau \vdash P = Q$ iff $P = Q$ can be derived from the axioms Ax, the τ-laws, and the usual rules for equality.

Theorem (Soundness) If $Ax_\tau \vdash P = Q$ then $P \cong Q$.

Properties of observation congruence

The following properties that can be useful to prove observation congruence

- If $P \approx Q$ then $\mu.P \cong \mu.Q$
- $P \approx Q$ iff $P \cong Q$ or $P \cong \tau.Q$ or $\tau.P \cong Q$

Exercise:

- Assume $A \stackrel{\text{def}}{=} a.b.0 + a.c.0$ and $B \stackrel{\text{def}}{=} a.(\tau.b.0 + \tau.c.0)$. Prove that $A \cong B$.
- Assume $A \stackrel{\text{def}}{=} (\nu b)(a.b.c.0 + a.\bar{b}.c.0)$ and $B \stackrel{\text{def}}{=} a.a.c.c.0$. Prove that $A \cong B$.

Two equivalent definitions of FIFO Queues

Consider the following definitions of a FIFO queue with 2 positions in value-passing CCS

First definition

$$\begin{array}{rcl} Q_2(in, out) & \stackrel{\text{def}}{=} & in(y).Q_1(in, out)\langle y\rangle \\ Q_1(in, out)\langle y\rangle & \stackrel{\text{def}}{=} & in(z).Q_0(in, out)\langle z, y\rangle \ + \ \overline{out}\langle y\rangle.Q_2(in, out) \\ Q_0(in, out)\langle z, y\rangle & \stackrel{\text{def}}{=} & \overline{out}\langle y\rangle.Q_1(in, out)\langle z\rangle \end{array}$$

Second definition

Here we define a 2-positions queue as the concatenation of two 1-position buffers

$$\begin{array}{rcl} B(in, out) & \stackrel{\text{def}}{=} & in(y).B'(in, out)\langle y\rangle \\ B'(in, out)\langle y\rangle & \stackrel{\text{def}}{=} & \overline{out}\langle y\rangle.B(in, out) \\ Queue(in, out) & \stackrel{\text{def}}{=} & (\nu c)(B(in, c) \mid B(c, out)) \end{array}$$

Exercise Prove that $Q_2(in, out)$ is observation-congruent to $Queue(in, out)$.