MPRI Concurrency (course number 2-3) 2005-2006: π-calculus

2006-02-15

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

James J. Leifer
INRIA Rocquencourt

James.Leifer@inria.fr

A summary of the π-calculus

- Core syntax
- Structural congruence ($\equiv$)
- Reduction ($\longrightarrow$)
- Labelled transitions ($\xrightarrow{\alpha}$)
- Strong bisimulation ($\sim$) and weak bisimulation ($\approx$)
- Strong barbs ($P\downarrow x$) and weak barbs ($P\Downarrow x$)
- "Up to" techniques (up to strong bisimilarity, up to contexts)

Features

- Sum ($\overline{x}y.P + \overline{w}z.Q$)
- Infinite behaviour ($!P$ or recursive definitions)
- Polyadic channels ($\overline{x}\vec{y}.P, \ldots$)

Core syntax

$$
\begin{array}{rll}
P ::= & \overline{x}y.P & \text{output} \\
& x(y).P & \text{input } (y \text{ binds in } P) \\
& \boldsymbol{\nu}x.P & \text{restriction (new) } (x \text{ binds in } P) \\
& P \mid P & \text{parallel (par)} \\
& \mathbf{0} & \text{empty}
\end{array}
$$

The free names of P are written $\mathsf{fn}(P)$.

$$
\begin{array}{lcl}
\mathsf{fn}(\overline{x}y.P) & = & \{x, y\} \cup \mathsf{fn}(P) \\
\mathsf{fn}(x(y).P) & = & \{x\} \cup (\mathsf{fn}(P) \setminus \{y\}) \\
\mathsf{fn}(\boldsymbol{\nu}x.P) & = & \mathsf{fn}(P) \setminus \{x\} \\
\mathsf{fn}(P \mid P') & = & \mathsf{fn}(P) \cup \mathsf{fn}(P') \\
\mathsf{fn}(\mathbf{0}) & = & \varnothing
\end{array}
$$

We consider processes up to alpha-conversion: provided $y' \notin \mathsf{fn}(P)$, we have

$$
\begin{array}{c}
x(y).P = x(y').\{y'/y\}P \\
\boldsymbol{\nu}y.P = \boldsymbol{\nu}y'.\{y'/y\}P
\end{array}
$$

Structural congruence ($\equiv$)

The smallest equivalence relation such that:

$$P \mid (Q \mid S) \equiv (P \mid Q) \mid S \qquad \text{(str-assoc)}$$

$$P \mid Q \equiv Q \mid P \qquad \text{(str-commut)}$$

$$P \mid \mathbf{0} \equiv P \qquad \text{(str-id)}$$

$$\boldsymbol{\nu} x.\boldsymbol{\nu} y.P \equiv \boldsymbol{\nu} y.\boldsymbol{\nu} x.P \qquad \text{(str-swap)}$$

$$\boldsymbol{\nu} x.\mathbf{0} \equiv \mathbf{0} \qquad \text{(str-zero)}$$

$$\boldsymbol{\nu} x.P \mid Q \equiv \boldsymbol{\nu} x.(P \mid Q) \qquad \text{if } x \notin \mathsf{fn}(Q) \qquad \text{(str-ex)}$$

And congruence rules:

$$\frac{P \equiv P'}{P \mid Q \equiv P' \mid Q} \; \text{(str-par-l)} \qquad \frac{P \equiv P'}{\boldsymbol{\nu} x.P \equiv \boldsymbol{\nu} x.P'} \; \text{(str-new)}$$

Note: we don’t close up by input or output prefixing.

Reduction ($\longrightarrow$)

We say that P reduces to P', written $P \longrightarrow P'$, if this can be derived from the following rules:

$$\overline{x}y.P \mid x(u).Q \longrightarrow P \mid \{y/u\}Q \qquad \text{(red-comm)}$$

$$\frac{P \longrightarrow P'}{P \mid Q \longrightarrow P' \mid Q} \qquad \text{(red-par)}$$

$$\frac{P \longrightarrow P'}{\boldsymbol{\nu} x.P \longrightarrow \boldsymbol{\nu} x.P'} \qquad \text{(red-new)}$$

We close reduction by structural congruence:

$$\frac{P \equiv\longrightarrow\equiv P'}{P \longrightarrow P'} \qquad \text{(red-str)}$$

Labels

The labels α are of the form:

$$\begin{array}{lll} \alpha ::= & \overline{x}y & \text{output} \\ & \overline{x}(y) & \text{bound output} \\ & xy & \text{input} \\ & \tau & \text{silent} \end{array}$$

The free names $\mathsf{fn}(\alpha)$ and bound names $\mathsf{bn}(\alpha)$ are defined as follows:

α	$\overline{x}y$	$\overline{x}(y)$	xy	τ
$\mathsf{fn}(\alpha)$	$\{x, y\}$	$\{x\}$	$\{x, y\}$	$\varnothing$
$\mathsf{bn}(\alpha)$	$\varnothing$	$\{y\}$	$\varnothing$	$\varnothing$

Labelled transitions ($P \xrightarrow{\alpha} P'$)

Labelled transitions are of the form $P \xrightarrow{\alpha} P'$ and are generated by:

$$\overline{x}y.P \xrightarrow{\overline{x}y} P \quad \text{(lab-out)} \qquad\qquad x(y).P \xrightarrow{xz} \{z/y\}P \quad \text{(lab-in)}$$

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \text{if } \mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \varnothing \quad \text{(lab-par-l)}$$

$$\frac{P \xrightarrow{\alpha} P'}{\boldsymbol{\nu} y.P \xrightarrow{\alpha} \boldsymbol{\nu} y.P'} \text{if } y \notin \mathsf{fn}(\alpha) \cup \mathsf{bn}(\alpha) \quad \text{(lab-new)} \qquad\qquad \frac{P \xrightarrow{\overline{x}y} P'}{\boldsymbol{\nu} y.P \xrightarrow{\overline{x}(y)} P'} \text{if } y \neq x \quad \text{(lab-open)}$$

$$\frac{P \xrightarrow{\overline{x}y} P' \qquad Q \xrightarrow{xy} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \quad \text{(lab-comm-l)} \qquad\qquad \frac{P \xrightarrow{\overline{x}(y)} P' \qquad Q \xrightarrow{xy} Q'}{P \mid Q \xrightarrow{\tau} \boldsymbol{\nu} y.(P' \mid Q')} \text{if } y \notin \mathsf{fn}(Q) \quad \text{(lab-close-l)}$$

plus symmetric rules (lab-par-r), (lab-comm-r), (lab-close-r).

Feature: sum

P	$::=$	M	sum
		$P \mid P$	parallel (par)
		$\boldsymbol{\nu} x.P$	restriction (new) (x binds in P)
M	$::=$	$\overline{x}y.P$	output
		$x(y).P$	input (y binds in P)
		$M + M$	sum
		$\mathbf{0}$	

Changes:

- structural congruence: $+$ is associative and commutative with identity $\mathbf{0}$.
- reduction: $(\overline{x}y.P + M) \mid (x(u).Q + N) \longrightarrow P \mid \{y/u\}Q$.
- labelled transition: $M + \overline{x}y.P + N \xrightarrow{\overline{x}y} P$
  $M + x(y).P + N \xrightarrow{xz} \{z/y\}P$

Feature: infinite behaviour via replication

Syntax: $P ::= \ldots !P$

Structural congruence: $!P \equiv P \mid !P$

Labelled transitions (easy to state):

$$\frac{P \mid !P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'} \text{if } \mathsf{bn}(\alpha) \cap \mathsf{fn}(P) = \varnothing \quad \text{(lab-bang)}$$

Labelled transitions (easy to use):

$$\frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P' \mid !P} \text{if } \mathsf{bn}(\alpha) \cap \mathsf{fn}(P) = \varnothing \quad \text{(lab-bang-simple)}$$

$$\frac{P \xrightarrow{\overline{x}y} P' \qquad P \xrightarrow{xy} P''}{!P \xrightarrow{\tau} (P' \mid P'') \mid !P} \quad \text{(lab-bang-comm)}$$

$$\frac{P \xrightarrow{\overline{x}(y)} P' \qquad P \xrightarrow{xy} P''}{!P \xrightarrow{\tau} \boldsymbol{\nu} y.(P' \mid P'') \mid !P} \text{if } y \notin \mathsf{fn}(P) \quad \text{(lab-bang-close)}$$

Feature: infinite behaviour via process abstraction

We can define a process abstractions:

$$F = (u_1, ..., u_k).P$$

Instantiation takes an abstraction and a vector of names and gives back a process:

$$F\langle x_1, ..., x_k \rangle = \{x_1/u_1, ..., x_k/u_k\}P$$

Feature: polyadic channels

In the syntax we extend our notion of *monadic* channels, which carry exactly one name, to *polyadic* channels, which carry a vector of names, i.e.

$$P ::= \overline{x}\langle y_1, ..., y_n \rangle.P \qquad \text{output}$$
$$\qquad x(y_1, ..., y_n).P \qquad \text{input } (y_1, ..., y_n \text{ pairwise distinct and bind in } P)$$

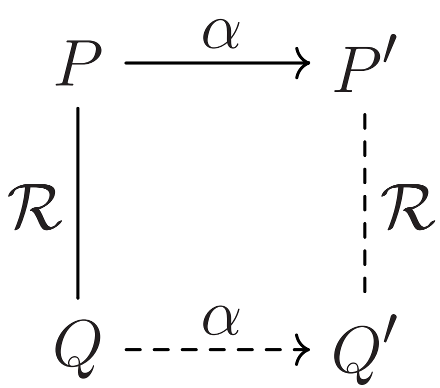
We then generalise the reduction rule as follows:

$$\overline{x}\vec{y}.P \mid x(\vec{u}).Q \longrightarrow P \mid \{\vec{y}/\vec{u}\}Q$$

(The label transitions become complicated because some of the elements of an output may be bound and some free.)

Strong bisimulation

A relation $\mathcal{R}$ is a strong bisimulation if it is symmetric and for all $(P, Q) \in \mathcal{R}$ and $P \xrightarrow{\alpha} P'$, where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \varnothing$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$.

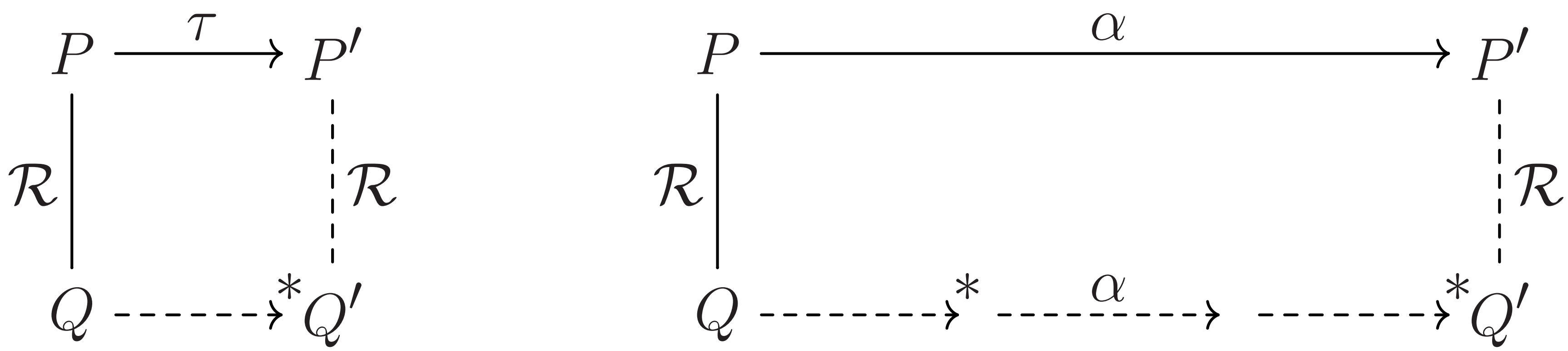


Strong bisimilarity $\sim$ is the largest strong bisimulation.

Weak bisimulation

A relation $\mathcal{R}$ is a weak bisimulation if it is symmetric and for all $(P,Q) \in \mathcal{R}$ and $P \xrightarrow{\alpha} P'$, where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \emptyset$, one of the following cases holds:

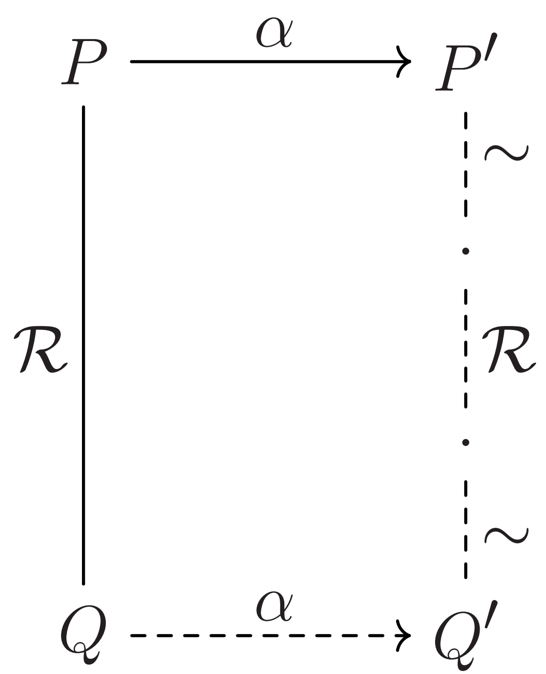
- If $\alpha = \tau$ then there exists Q' such that $Q \longrightarrow^* Q'$ and $(P',Q') \in \mathcal{R}$.
- If $\alpha \neq \tau$ then there exists Q' such that $Q \longrightarrow^* \xrightarrow{\alpha} \longrightarrow^* Q'$ and $(P',Q') \in \mathcal{R}$.



Weak bisimilarity $\approx$ is the largest weak bisimulation.

Strong bisimulation up to strong bisimilarity

Suppose for all $(P, Q) \in \mathcal{R}$ and $P \xrightarrow{\alpha} P'$, where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \varnothing$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in {\sim}\mathcal{R}{\sim}$, and symmetrically.



Then ${\sim}\mathcal{R}{\sim}$ is a strong bisimulation. Is $\mathcal{R}$ also a strong bisimulation?

Evaluation contexts

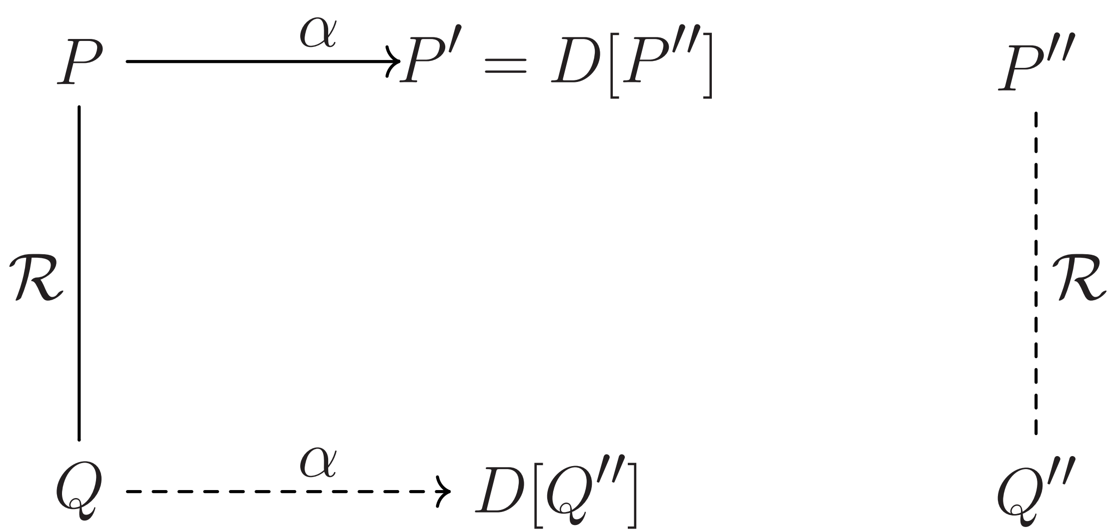
Let $\mathcal{E}$ be the set of evaluation contexts; these are generated by the grammar:

$$
\begin{aligned}
D \in \mathcal{E} ::= \ & \_ \\
& D \mid P \\
& P \mid D \\
& \boldsymbol{\nu} x.D
\end{aligned}
$$

What isn't an evaluation context?

Strong bisimulation up to contexts

Suppose for all $(P, Q) \in \mathcal{R}$ and $P \xrightarrow{\alpha} P'$, where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \varnothing$, there exists $D \in \mathcal{E}$, P'', and Q'' such that $P' = D[P'']$ and $Q \xrightarrow{\alpha} D[Q'']$ and $(P'', Q'') \in \mathcal{R}$, and symmetrically.



Then $\{(D[P], D[Q]) \;/\; (P, Q) \in \mathcal{R}, D \in \mathcal{E}\}$ is a strong bisimulation.

Example: $!!P \sim !P$.

Barbs

A process P has a strong barb x, written $P{\downarrow}x$ iff there exists P_0, P_1, and $\vec{y}$ such that $P \equiv \boldsymbol{\nu}\vec{y}.(\overline{x}u.P_0 \mid P_1)$ and $x \notin \vec{y}$.

A process P has a weak barb x, written $P{\Downarrow}x$ iff there exists P' such that $P \longrightarrow^{*} P'$ and $P'{\downarrow}x$.