Concurrency 8

The π-calculus

Catuscia Palamidessi
catuscia@lix.polytechnique.fr

Note: the material of today's lecture is partly from:

R. Milner, J. Parrow, D. Walker. Modal Logics for Mobile Processes. *Theoretical Computer Science* **114**:149-171, 1993. http://www.sics.se/~joachim/modmob.ps.Z

The π-calculus: syntax

- **Similar to CCS with value passing, but values are channel names, and recursion is replaced by replication (!)**

$$\pi \;::=\; x(y) \mid \bar{x}y \mid \tau$$

action prefixes (input, output, silent)
x, y are channel names

$$\begin{array}{lll} P & ::= & O \\ & \mid & \pi.P \\ & \mid & P \mid P \\ & \mid & P + P \\ & \mid & (\nu x)P \\ & \mid & !\,P \end{array}$$

inaction
prefix
parallel
sum
restriction, new name
replication

The π-calculus: syntax

- Names: $n(P)$
  - Free $fn(P)$
  - Bound $bn(P)$
    - Input and restriction are binders
  - Exercise: give the formal definition of $fn(P)$ and $bn(P)$

Example: $P = ((\nu x)\bar{y}x.x(z).\bar{z}x.0) \mid (y(w).\bar{w}u.0)$

we have: $fn(P) = \{y, u\}$, $bn(P) = \{x, z, w\}$

- Alpha conversion

Example: $Q = ((\nu v)\bar{y}v.v(z).\bar{z}v.0) \mid (y(x).\bar{x}u.0)$

we have: $P \equiv_\alpha Q$

The π-calculus: structural equivalence

- Introduced to simplify the description of the operational semantics
  - If $P \equiv_\alpha Q$ then $P \equiv Q$
  - $P \mid Q \equiv Q \mid P$
  - $P + Q \equiv Q + P$
  - $!\,P \equiv P \mid !\,P$

- Some presentations include other equivalences, for instance:
  - $P \mid 0 \equiv P$, $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$
  - $P + 0 \equiv P$, $(P + Q) + R \equiv P + (Q + R)$, $P + P \equiv P$
  - $(\nu\, x)\,(\nu\, y)\,P \equiv (\nu\, y)\,(\nu x)\,P$, $(\nu\, x)\,P \equiv P$ if $\neg\, x \in fn(P)$
  - $P \mid (\nu\, x)\,Q \equiv (\nu\, x)\,(P \mid Q)$ if $\neg\, x \in fn(P)$ (scope extrusion)

The π–calculus: operational semantics

- The early operational semantics of the π-calculus is defined as a labeled transition system. Transitions have the form

$$P \xrightarrow{\mu} Q$$

Here P and Q are processes and μ is an action, where actions are defined as follows:

μ	kind	$fn(\mu)$	$bn(\mu)$
τ	silent	$\emptyset$	$\emptyset$
$x(y)$	input	$\{x\}$	$\{y\}$
$\bar{x}y$	free output	$\{x, y\}$	$\emptyset$
$\bar{x}(y)$	bound output	$\{x\}$	$\{y\}$

The π–calculus: early operational semantics

E-Input $\dfrac{}{x(y).P \xrightarrow{x(z)} P\{z/y\}}$

Output/τ $\dfrac{}{\alpha.P \xrightarrow{\alpha} P}$ $\alpha = \bar{x}y$ or $\alpha = \tau$

Par $\dfrac{P \xrightarrow{\mu} P'}{P \mid Q \xrightarrow{\mu} P' \mid Q}$ $bn(\mu) \cap fn(Q) = \emptyset$

Sum $\dfrac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P'}$

Res $\dfrac{P \xrightarrow{\mu} P'}{\nu y P \xrightarrow{\mu} \nu y P'}$ $y \notin n(\mu)$

Open $\dfrac{P \xrightarrow{\bar{x}y} P'}{\nu y P \xrightarrow{\bar{x}(y)} P'}$ $x \neq y$

E-Com $\dfrac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\bar{x}y} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$

Close $\dfrac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\bar{x}(y)} Q'}{P|Q \xrightarrow{\tau} \nu y(P'|Q')}$ $y \notin fn(P)$

Cong $\dfrac{P' \equiv P \qquad P \xrightarrow{\mu} Q \qquad Q \equiv Q'}{P' \xrightarrow{\mu} Q'}$

The π–calculus: early bisimulation

Definition

- We say that a binary relation $\mathcal{S}$ is an early simulation if $P\,\mathcal{S}\,Q$ implies that

  if $P \xrightarrow{\mu} P'$ and μ is any action with $bn(\mu) \cap fn(P, Q) = \emptyset$, then for some Q', $Q \xrightarrow{\mu} Q'$ and $P'\,\mathcal{S}\,Q'$.

- The relation $\mathcal{S}$ is an early bisimulation iff both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are early simulations.

- P and Q are early bisimilar, notation $P \sim_E Q$, iff $P\,\mathcal{S}\,Q$ for some early bisimulation $\mathcal{S}$.

- **Question:** is $\sim_E$ a congruence?
  Answer: **No.** Example:

  $$x(z).0 \mid \bar{y}z.0 \sim_E x(z).\bar{y}z.0 + \bar{y}z.x(z).0$$

  but

  $$w(x)(x(z).0 \mid \bar{y}z.0) \not\sim_E w(x)(x(z).\bar{y}z.0 + \bar{y}z.x(z).0)$$

The π–calculus: late semantics

- The late operational semantics is obtained by replacing rules E-Input and E-Com by the following:

$$\text{L-Input} \quad \frac{}{x(y).P \xrightarrow{x(y)} P} \qquad\qquad \text{L-Com} \quad \frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\bar{x}z} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'}$$

- **Definition** We say that a binary relation $\mathcal{S}$ is a late simulation if $P\,\mathcal{S}\,Q$ implies that

  1. if $P \xrightarrow{\mu} P'$ and μ is τ or output, with $bn(\mu) \cap fn(P,Q) = \emptyset$, then for some Q', $Q \xrightarrow{\mu} Q'$ and $P'\,\mathcal{S}\,Q'$.
  2. if $P \xrightarrow{x(y)} P'$ and $y \notin fn(P,Q) = \emptyset$, then for some Q', $Q \xrightarrow{x(y)} Q'$ and for all z, $P'\{z/y\}\,\mathcal{S}\,Q'\{z/y\}$.

- The relation $\mathcal{S}$ is a late bisimulation iff both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are late simulations.

- P and Q are late bisimilar, notation $P \sim_L Q$, iff $P\,\mathcal{S}\,Q$ for some late bisimulation $\mathcal{S}$.

Late vs early bisimulation

Late bisimulation is strictly more discriminating than early bisimulation.

Example

$P \equiv x(y).R + x(y).S$

$Q \equiv x(y).R + x(y).S + x(y).$ if $y = z$ then R else S

We have that $P \sim_E Q$ but $P \not\sim_L Q$

In order to obtain a process that behave like the if-then-else we can use synchronization:

$Q \equiv x(y).R + x(y).S + (x(y).(\bar{y}w.0 \mid z(w).0))$

where $R \equiv \bar{z}w.0 \mid z(w).0$ and $S \equiv \bar{y}w.z(w).0 + z(w).\bar{y}w.0$, while P is defined as before.