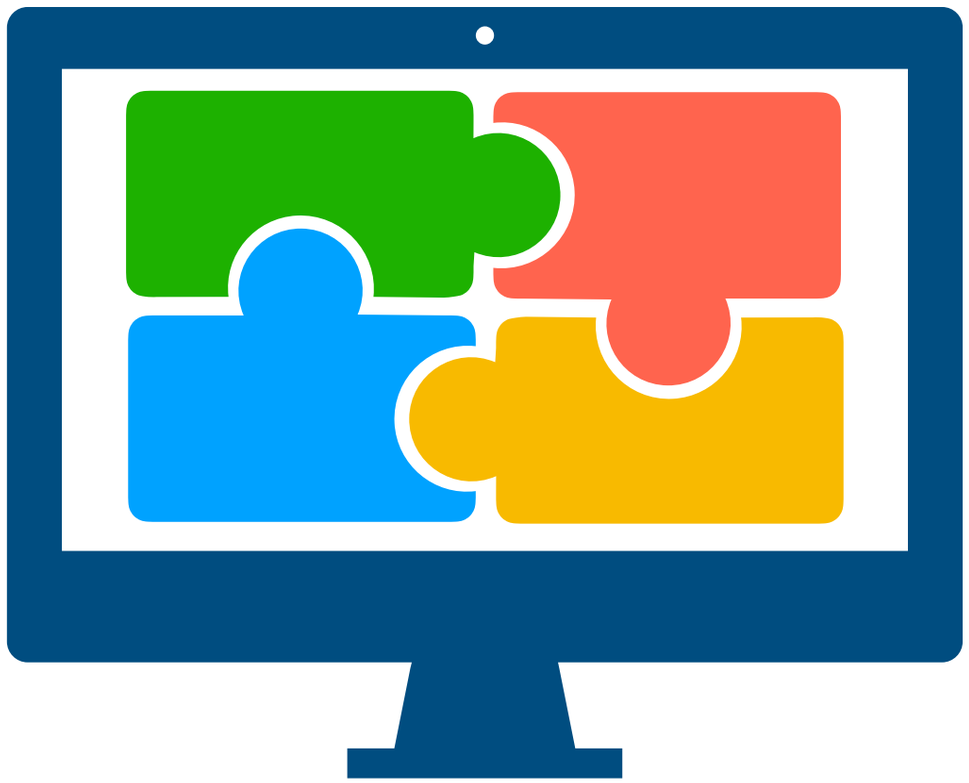


http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/

PSC 2020/21 (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni
http://www.di.unipi.it/~bruni/

25 - Pi-calculus

π-calculus

Name mobility

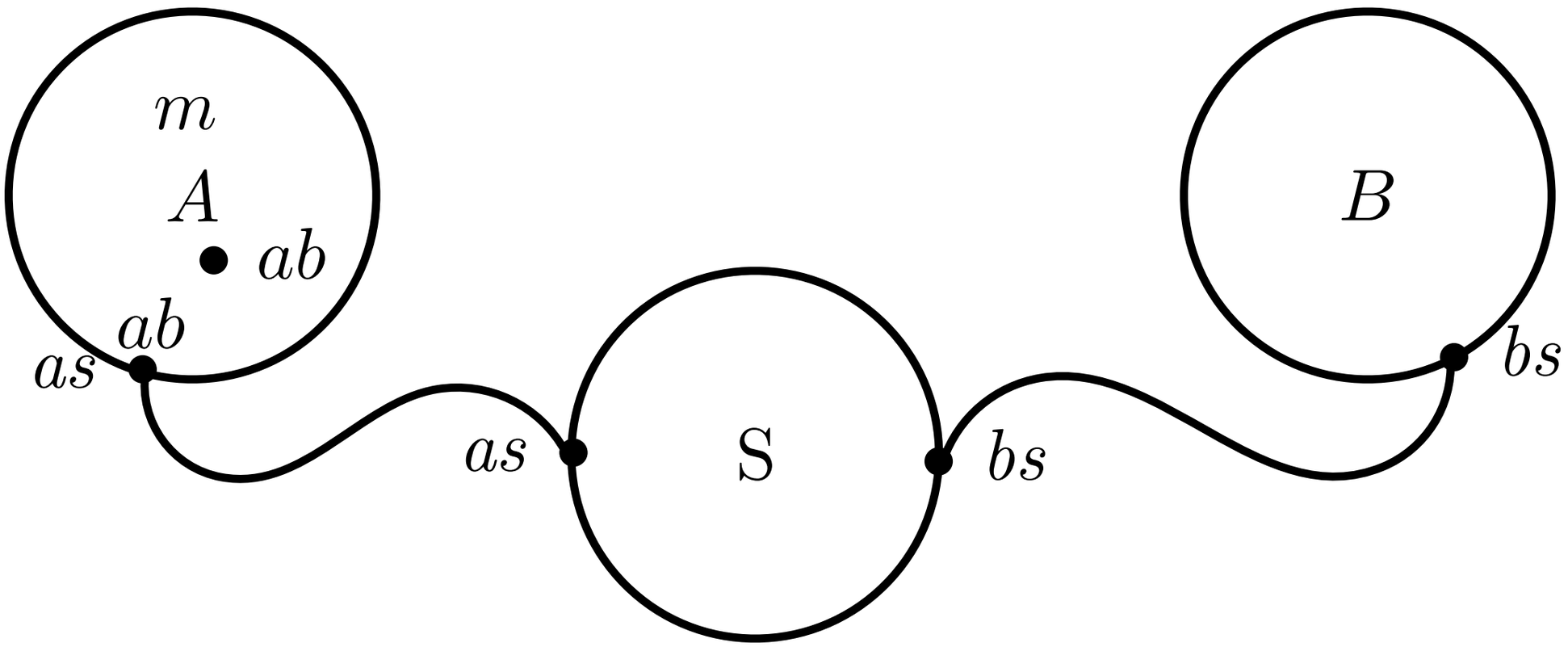
in CCS communication is statically defined

(channels cannot be transmitted / received)

what if channels are treated as ordinary values?

can we extend CCS to communicate communication means?

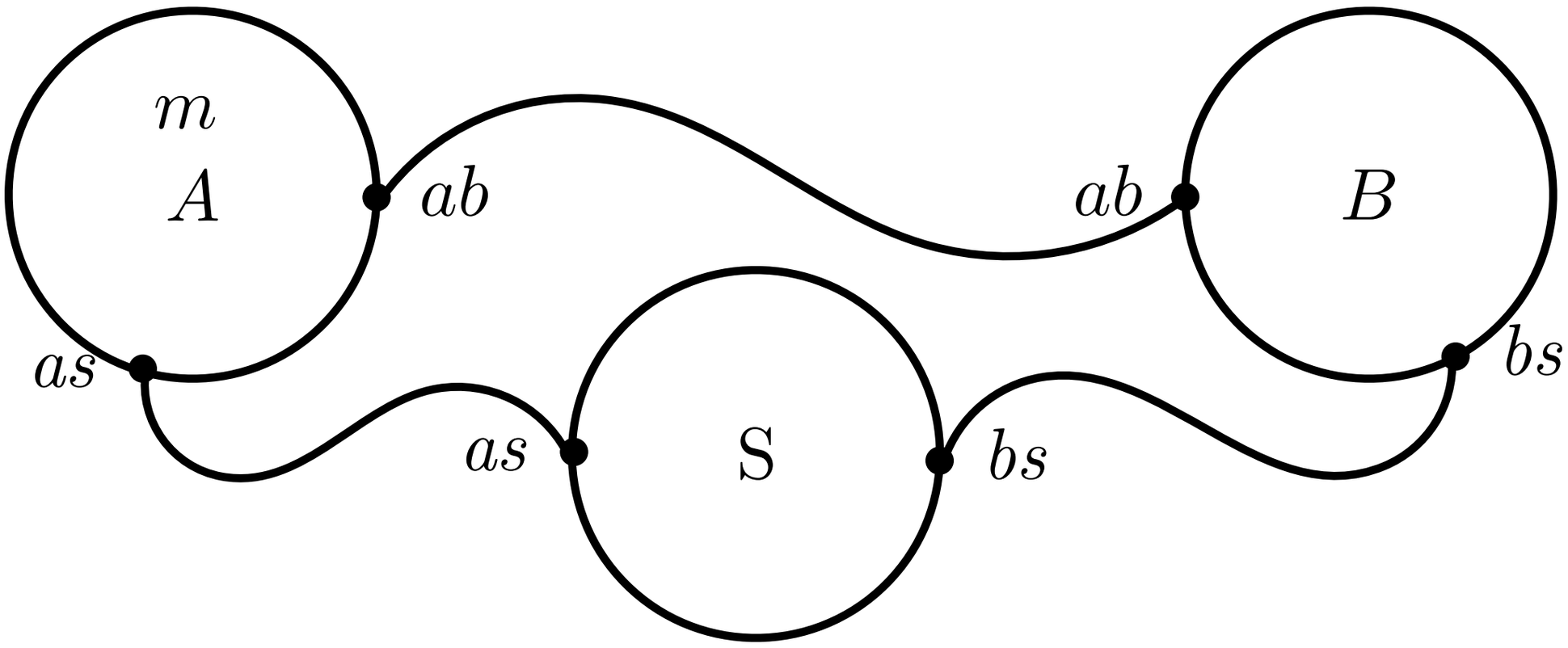
Name mobility



Name mobility

$(A\backslash ab \mid B \mid S)\backslash as\backslash bs$

$((A \mid B)\backslash ab \mid S)\backslash as\backslash bs$



pi-calculus: syntax

π	$::=$	$\overline{x}y$	send *y* on *x*
	$\mid$	$x(y)$	receive on *x* and store in *y*
	$\mid$	τ	internal action

p	$::=$	$\mathbf{nil}$	inactive process
	$\mid$	$\pi.p$	action prefix
	$\mid$	$[x = y]p$	matching (like, if *x*=*y* then *p*)
	$\mid$	$p + p$	nondeterministic choice
	$\mid$	$p \mid p$	parallel composition
	$\mid$	$(y)p$	restriction (like $p \backslash y$, also written $(\nu y)p$)
	$\mid$	$!p$	replication (like $p \mid p \mid p \mid \cdots$)

pi-calculus: syntax

the only binders are input prefix and name restriction

$$x(y).p \qquad\qquad (y)p$$

the notion of free and bound names are defined as usual

$$\mathsf{fn}(p) \qquad\qquad \mathsf{bn}(p)$$

the notion of capture avoiding substitution is defined as usual

$$p[^{y}/_{x}]$$

processes are taken up-to alpha-renaming of bound names

$$x(y).p = x(z).(p[^{z}/_{y}]) \text{ if } z \notin \mathsf{fn}(p)$$

$$(y)p = (z)(p[^{z}/_{y}]) \text{ if } z \notin \mathsf{fn}(p)$$

pi-calculus: notation

write $\sum_{i=1}^{n} p_i$ instead of $p_1 + \cdots + p_n$

write $\prod_{i=1}^{n} p_i$ instead of $p_1 | \cdots | p_n$

write $(\widetilde{a})p$ instead of $(a_1) \cdots (a_n)p$

input guarded replication: $!x(y).p$

pi-calculus: conventions

we omit trailing **nil**

$$(y)\overline{x}y.\overline{y}m.\mathbf{nil} \quad \text{as} \quad (y)\overline{x}y.\overline{y}m$$

vacuous messages written as CCS prefixes

$$\overline{x} \mid x.p + x.q$$

mismatch not necessary (if the number of cases is finite)

$$[x \neq y_i].p \quad \text{as} \quad \sum_{j \neq i} [x = y_j]p$$

polyadic communication (can be encoded)

$$\overline{x}\langle z_1, ..., z_n \rangle.p \qquad x(y_1, ..., y_n).q$$

$$\overline{x}\langle \widetilde{z} \rangle.p \qquad x(\widetilde{y}).q$$

Examples

$A \triangleq (ab)\,\overline{as}\,ab\,.\,\overline{ab}\,m\,.\,A'$

do something else

send *m* on *ab*

send *ab* on *as*

create a new (private) channel *ab*

$S \triangleq !\,as(x)\,.\,\overline{bs}\,x$

omit **nil**

receive on *as* and forward to on *bs*

$B \triangleq \; bs(y)\,.\,y(w)\,.\,B'$

do something else

receive on *y*

receive *y* on *bs*

$(as)\,(bs)\,(A \mid S \mid B)$

Example

$$(as)(bs)(\; bs(y)\,.\,y(w)\,.\,B' \;\mid\; !\boxed{as(x)}.\,\overline{bs}\,x \;\mid\; (ab)\boxed{\overline{as}\,ab}.\,\overline{ab}\,m\,.\,A'\;)$$

scope extrusion

τ

$$(as)(bs)(\boxed{bs(y)}\,.\,y(w)\,.\,B' \;\mid\; (ab)(\;\boxed{\overline{bs}\,ab}\mid \qquad \overline{ab}\,m\,.\,A'\;)\mid S\;)$$

scope extrusion

τ

$$(as)(bs)(\;(ab)(\boxed{ab(w)}.\,B' \;\mid\; \mathbf{nil} \;\mid\; \boxed{\overline{ab}\,m}.\,A'\;)\mid S\;)$$

dynamic topology of communication

π-calculus operational semantics

pi-calculus: LTS

ongoing interaction
with the environment
(with other processes)

$$p \xrightarrow{\alpha} q$$

a process
in its current
state

the process
state after the
interaction

small-step semantics

pi-calculus labels

What is a label α?

$$
\begin{array}{lll}
\alpha & ::= & \tau \\
 & | & \overline{x}y \\
 & | & x(y) \\
 & | & \ldots
\end{array}
$$

free names

bound name

Action prefix

$$\text{Act)}\ \frac{}{\pi.p \xrightarrow{\pi} p}$$

note: $x(y).p \xrightarrow{x(y)} p$

$$x(y).p = x(z).(p[^z/_y]) \text{ if } z \notin \mathsf{fn}(p)$$

$$x(y).p \xrightarrow{x(z)} p[^z/_y] \text{ if } z \notin \mathsf{fn}(p)$$

we call *fresh* such *z*

z is just a placeholder, not an actual value

we define some sort of symbolic semantics for input
(the continuation needs an actual name to evolve)

Matching & Choice

$$\text{Match)}\ \frac{p \xrightarrow{\alpha} q}{[x = x]p \xrightarrow{\alpha} q}$$

example: $login(y)\,.\,[y = pwd]\,p$

$$\text{SumL)}\ \frac{p_1 \xrightarrow{\alpha} q}{p_1 + p_2 \xrightarrow{\alpha} q} \qquad \text{SumR)}\ \frac{p_2 \xrightarrow{\alpha} q}{p_1 + p_2 \xrightarrow{\alpha} q}$$

example: $confirm(y)\,.\,(\,[y = ok]\,p + \,[y = no]\,q\,)$

Parallel

$$\text{ParL)}\ \frac{p_1 \xrightarrow{\alpha} q_1}{p_1 | p_2 \xrightarrow{\alpha} q_1 | p_2}\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(p_2) = \emptyset$$

$$\text{ParR)}\ \frac{p_2 \xrightarrow{\alpha} q_2}{p_1 | p_2 \xrightarrow{\alpha} p_1 | q_2}\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(p_1) = \emptyset$$

fresh

$$x(y) \,.\, \overline{y}z \mid w(u) \xrightarrow{x(v)} \overline{v}z \mid w(u)$$

not fresh!

not taken in input!

$$x(y) \,.\, \overline{y}z \mid w(u) \xrightarrow{x(w)} \overline{w}z \mid w(u)$$

Communication

ComL) $$\frac{p_1 \xrightarrow{\overline{x}z} q_1 \quad p_2 \xrightarrow{x(y)} q_2}{p_1|p_2 \xrightarrow{\tau} q_1|(q_2[^z/_y])}$$

ComR) $$\frac{p_1 \xrightarrow{x(y)} q_1 \quad p_2 \xrightarrow{\overline{x}z} q_2}{p_1|p_2 \xrightarrow{\tau} q_1[^z/_y]|q_2}$$

$$\overline{x}z\,.\,y(v) \mid x(y)\,.\,\overline{y}z \xrightarrow{\tau} y(v) \mid \overline{z}z$$

$\overline{x}z$ ↓ $\qquad$ ↓ $x(y)$

$y(v) \qquad\qquad \overline{y}z$

Restriction

$$\text{Res)}\ \frac{p \xrightarrow{\alpha} q}{(y)p \xrightarrow{\alpha} (y)q}\ y \notin \mathsf{fn}(\alpha) \cup \mathsf{bn}(\alpha)$$

$$\text{Open)}\ \frac{p \xrightarrow{\overline{x}y} q}{(y)p \xrightarrow{\overline{x}(z)} q[^z/_y]}\quad \begin{array}{l} y \neq x \\ z \notin \mathsf{fn}((y)p) \end{array}$$

a new kind of label:
bound output

we temporarily remove the restriction

$$\begin{array}{lcl} \alpha & ::= & \tau \\ & | & \overline{x}y \\ & | & x(y) \\ & | & \overline{x}(y) \end{array}$$

$$(y)\overline{x}y \,.\, \overline{y}w \xrightarrow{\overline{x}(y)} \overline{y}w$$

Scope extrusion

CloseL) $$\frac{p_1 \xrightarrow{\overline{x}(z)} q_1 \quad p_2 \xrightarrow{x(z)} q_2}{p_1 | p_2 \xrightarrow{\tau} (z)(q_1 | q_2)}$$

CloseR) $$\frac{p_1 \xrightarrow{x(z)} q_1 \quad p_2 \xrightarrow{\overline{x}(z)} q_2}{p_1 | p_2 \xrightarrow{\tau} (z)(q_1 | q_2)}$$

we restore the restriction

$$x(v) \,.\, v(b) \mid (y)\, \overline{x}y \,.\, \overline{y}a \xrightarrow{\tau} (z)(z(b) \mid \overline{z}a)$$

$x(z)$ ↓ $z(b)$; $\overline{x}(z)$ ↓ $\overline{z}a$

Replication

$$\text{Rep)}\ \frac{p|!p \xrightarrow{\alpha} q}{!p \xrightarrow{\alpha} q}$$

alternatively (to avoid infinite branching):

$$\text{Rep)}\ \frac{p|p \xrightarrow{\alpha} q}{!p \xrightarrow{\alpha} q|!p}$$

alternatively (if only input guarded replication is allowed):

$$\text{Rep)}\ \frac{p \xrightarrow{\alpha} q}{!p \xrightarrow{\alpha} q|!p}$$

$$!req(y).p \xrightarrow{req(y)} p|!req(y).p$$

pi-calculus: semantics

Act) $\dfrac{}{\pi.p \xrightarrow{\pi} p}$

Match) $\dfrac{p \xrightarrow{\alpha} q}{[x = x]p \xrightarrow{\alpha} q}$

SumL) $\dfrac{p_1 \xrightarrow{\alpha} q}{p_1 + p_2 \xrightarrow{\alpha} q}$ SumR)…

Rep) $\dfrac{p|!p \xrightarrow{\alpha} q}{!p \xrightarrow{\alpha} q}$

ParL) $\dfrac{p_1 \xrightarrow{\alpha} q_1}{p_1|p_2 \xrightarrow{\alpha} q_1|p_2}$ $\mathsf{bn}(\alpha) \cap \mathsf{fn}(p_2) = \emptyset$ ParR)…

Res) $\dfrac{p \xrightarrow{\alpha} q}{(y)p \xrightarrow{\alpha} (y)q}$ $y \notin \mathsf{fn}(\alpha) \cup \mathsf{bn}(\alpha)$

ComL) $\dfrac{p_1 \xrightarrow{\overline{x}z} q_1 \quad p_2 \xrightarrow{x(y)} q_2}{p_1|p_2 \xrightarrow{\tau} q_1|(q_2[{}^{z}/_{y}])}$ ComR)…

Open) $\dfrac{p \xrightarrow{\overline{x}y} q}{(y)p \xrightarrow{\overline{x}(z)} q[{}^{z}/_{y}]}$ $y \neq x$, $z \notin \mathsf{fn}((y)p)$

CloseL) $\dfrac{p_1 \xrightarrow{\overline{x}(z)} q_1 \quad p_2 \xrightarrow{x(z)} q_2}{p_1|p_2 \xrightarrow{\tau} (z)(q_1|q_2)}$ CloseR)…

Process constants

$$\Delta = \left\{ \begin{array}{rcl} A_1(\widetilde{x}_1) & \triangleq & p_1 \\ \cdots & & \\ A_n(\widetilde{x}_n) & \triangleq & p_n \end{array} \right\} \qquad \begin{array}{rcll} p & ::= & \cdots & \\ & | & A\langle \widetilde{y} \rangle & \text{invocation} \end{array}$$

$$\frac{A(\widetilde{x}) \triangleq p \in \Delta \quad p[\widetilde{y}/\widetilde{x}] \xrightarrow{\alpha} q}{A\langle \widetilde{y} \rangle \xrightarrow{\alpha} q}$$

example $\quad A(x) \triangleq x(y).B\langle y \rangle \qquad B(y) \triangleq (z)\overline{y}z.A\langle z \rangle$

$$\begin{aligned} A\langle x \rangle \mid B\langle x \rangle &\xrightarrow{\tau} (z_1)(B\langle z_1 \rangle \mid A\langle z_1 \rangle) \\ &\xrightarrow{\tau} (z_1)(z_2)(A\langle z_2 \rangle \mid B\langle z_2 \rangle) \\ &\xrightarrow{\tau} (z_1)(z_2)(z_3)(B\langle z_3 \rangle \mid A\langle z_3 \rangle) \end{aligned}$$

Constants vs replication

$$\Delta = \left\{ \begin{array}{rcl} A_1(\widetilde{x}_1) & \triangleq & p_1 \\ \cdots & & \\ A_n(\widetilde{x}_n) & \triangleq & p_n \end{array} \right\} \qquad \begin{array}{rcll} p & ::= & \cdots & \\ & | & A\langle \widetilde{y} \rangle & \text{invocation} \end{array}$$

introduce one dedicated channel a_i for each constant A_i

$$D \triangleq \prod_{i=1}^{n} !a_i(\widetilde{x}_i).\widehat{p}_i$$ (requires polyadic communication)

where $\widehat{p} \triangleq p[{}^{\overline{a}_1}/_{A_1}, \ldots, {}^{\overline{a}_n}/_{A_n}]$

then run $(\widetilde{a})(\widehat{q} \mid D)$ instead of q

Example

$$A(x) \triangleq x(y).B\langle y\rangle \qquad\qquad B(y) \triangleq (z)\overline{y}z.A\langle z\rangle$$

$$D \triangleq \prod_{i=1}^{n} !a_i(\widetilde{x}_i).\widehat{p}_i \qquad \text{where } \widehat{p} \triangleq p[{}^{\overline{a}_1}/_{A_1}, \ldots, {}^{\overline{a}_n}/_{A_n}]$$

$$D \triangleq !a(x).x(y).\overline{b}y \mid !b(y).(z)\overline{y}z.\overline{a}z$$

then run $(\widetilde{a})(\widehat{q} \mid D)$ instead of q $\qquad\qquad q = A\langle x\rangle \mid B\langle x\rangle$

$$\begin{aligned}(a,b)(\overline{a}x \mid \overline{b}x \mid D) &\xrightarrow{\tau} (a,b)(x(y).\overline{b}y \mid \overline{b}x \mid D)\\ &\xrightarrow{\tau} (a,b)(x(y).\overline{b}y \mid (z)\overline{x}z.\overline{a}z \mid D)\\ &\xrightarrow{\tau} (a,b,z_1)(\overline{b}z_1 \mid \overline{a}z_1 \mid D)\end{aligned}$$

Exercise

what's wrong with the following encoding of matching?

$$[x = y]p \quad \text{as} \quad \overline{x} \mid y.p$$

there can be interferences!

other processes can steal the message on *x*

other processes can send messages on *y*

π-calculus bisimulation

Strong early ground bis

$$\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$$

$$p\,\mathbf{R}\,q \Rightarrow \begin{cases} p \xrightarrow{\tau} p' & \Rightarrow \exists q'.\ q \xrightarrow{\tau} q' \wedge p'\,\mathbf{R}\,q' \\ p \xrightarrow{\overline{x}y} p' & \Rightarrow \exists q'.\ q \xrightarrow{\overline{x}y} q' \wedge p'\,\mathbf{R}\,q' \\ p \xrightarrow{\overline{x}(y)} p' \quad y \notin \mathsf{fn}(q) & \Rightarrow \exists q'.\ q \xrightarrow{\overline{x}(y)} q' \wedge p'\,\mathbf{R}\,q' \\ p \xrightarrow{x(y)} p' \quad y \notin \mathsf{fn}(q) & \Rightarrow \forall z.\ \exists q'.\ q \xrightarrow{x(y)} q' \wedge p'[^{z}/_{y}]\,\mathbf{R}\,q'[^{z}/_{y}] \\ \text{and vice versa} & \end{cases}$$

for every possible input

Strong early ground bis

$$\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$$

$p \,\mathbf{R}\, q \Rightarrow$

- $p \xrightarrow{\alpha} p'$, $\mathsf{bn}(\alpha) \cap \mathsf{fn}(q) = \emptyset \;\Rightarrow \exists q'.\; q \xrightarrow{\alpha} q' \wedge p' \,\mathbf{R}\, q'$, $\alpha \neq x(y)$
- $p \xrightarrow{x(y)} p'$, $y \notin \mathsf{fn}(q) \;\Rightarrow \forall z.\; \exists q'.\; q \xrightarrow{x(y)} q' \wedge p'[^z/_y] \,\mathbf{R}\, q'[^z/_y]$

and vice versa

for every possible input

Strong early ground bis

$$\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$$

$$p\,\mathbf{R}\,q \Rightarrow \begin{cases} p \xrightarrow{\alpha} p' \\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(q) = \emptyset \;\; \Rightarrow \exists q'.\; q \xrightarrow{\alpha} q' \wedge p'\,\mathbf{R}\,q' \\ \alpha \neq x(y) \end{cases}$$

$$\begin{cases} p \xrightarrow{x(y)} p' \\ y \notin \mathsf{fn}(q) \end{cases} \Rightarrow \boxed{\forall z.}\; \exists q'.\; q \xrightarrow{x(y)} q' \wedge p'[^z/_y]\,\mathbf{R}\,q'[^z/_y]$$

and vice versa

early quantification

Strong late ground bis

$$\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$$

$$p\,\mathbf{R}\,q \Rightarrow \begin{cases} p \xrightarrow{\alpha} p' \\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(q) = \emptyset \;\; \Rightarrow \exists q'.\; q \xrightarrow{\alpha} q' \wedge p'\,\mathbf{R}\,q' \\ \alpha \neq x(y) \end{cases}$$

$$\begin{cases} p \xrightarrow{x(y)} p' \\ y \notin \mathsf{fn}(q) \end{cases} \Rightarrow \exists q'.\; q \xrightarrow{x(y)} q' \wedge \forall z.\; p'[^z/_y]\,\mathbf{R}\,q'[^z/_y]$$

and vice versa

late quantification

Early vs late

early for every input Bob can choose a different move

$$\forall w.\, \exists q'.\ \cdots$$

late Bob must choose one move, valid for every input

$$\exists q'.\, \forall w.\ \cdots$$

Early/late bisimilarities

early bisimilarity

$p \sim_E q$ iff $\exists$ a strong early ground bisimulation $\mathbf{R}$ with $p \, \mathbf{R} \, q$

late bisimilarity

$p \sim_L q$ iff $\exists$ a strong late ground bisimulation $\mathbf{R}$ with $p \, \mathbf{R} \, q$

Early vs late

$$P \triangleq x(y)\,.\,\tau + x(y)$$

$$P \xrightarrow{x(z)} \tau \xrightarrow{\tau} \mathbf{nil}$$

$$P \xrightarrow{x(z)} \mathbf{nil}$$

z fresh

$$Q \triangleq P + x(y)\,.\,[y = a]\tau$$

$$Q \xrightarrow{x(z)} \tau \xrightarrow{\tau} \mathbf{nil}$$

$$Q \xrightarrow{x(z)} \mathbf{nil}$$

$$Q \xrightarrow{x(z)} [z = a]\tau$$

z fresh

early game

Alice picks $Q \xrightarrow{x(z)} [z = a]\tau$

for $z = a$ Bob picks $P \xrightarrow{x(z)} \tau$

for $z \neq a$ Bob picks $P \xrightarrow{x(z)} \mathbf{nil}$

Bob wins

$$P \sim_E Q$$

Early vs late

$P \triangleq x(y) \,.\, \tau + x(y)$

$P \xrightarrow{x(z)} \tau \xrightarrow{\tau} \mathbf{nil}$

$P \xrightarrow{x(z)} \mathbf{nil}$

z fresh

$Q \triangleq P + x(y) \,.\, [y = a]\tau$

$Q \xrightarrow{x(z)} \tau \xrightarrow{\tau} \mathbf{nil}$

$Q \xrightarrow{x(z)} \mathbf{nil}$

$Q \xrightarrow{x(z)} [z = a]\tau$

z fresh

late game

Alice picks $Q \xrightarrow{x(z)} [z = a]\tau$

if Bob picks $P \xrightarrow{x(z)} \tau$ Alice wins when $z \neq a$

if Bob picks $P \xrightarrow{x(z)} \mathbf{nil}$ Alice wins when $z = a$

Alice wins

$$P \not\sim_L Q$$

Early vs late: properties

$$P \sim_L Q \Rightarrow P \sim_E Q$$

$$P \sim_E Q \not\Rightarrow P \sim_L Q$$

(see previous example)

$\sim_E$ (resp. $\sim_L$) is a strong early (resp. late) ground bisimulation

$\sim_E$ and $\sim_L$ are equivalences but not congruences

(not preserved by input prefix)

$p \simeq_E q$ iff $p\sigma \sim_E q\sigma$ for every substitution σ

$p \simeq_L q$ iff $p\sigma \sim_L q\sigma$ for every substitution σ

$\simeq_E$ and $\simeq_L$ are congruences

(the largest congruences included in each bisimilarity)

Early & late: properties

$$p + \mathbf{nil} \simeq p$$

$$p + q \simeq q + p$$

$$p + (q + r) \simeq (p + q) + r$$

$$p + p \simeq p$$

$$p|\mathbf{nil} \simeq p$$

$$p|q \simeq q|p$$

$$p|(q|r) \simeq (p|q)|r$$

$$(x)\mathbf{nil} \simeq \mathbf{nil}$$

$$(x)(y)p \simeq (y)(x)p$$

$$(x)(x)p \simeq (x)p$$

$$(x)(p + q) \simeq (x)p + (x)q$$

$$(x)(p|q) \simeq p|(x)q \text{ if } x \notin \mathsf{fn}(p)$$

$$[x = x]p \simeq p$$

$$!p \simeq p|!p$$

Weak semantics

weak transitions: $p \stackrel{\alpha}{\Rightarrow} q$ defined as expected

weak bisimulations: Bob replies with weak moves

weak bisimilarities: $\approx_E$ and $\approx_L$ defined as expected

$\approx_E$ and $\approx_L$ are weak bisimulations

$\approx_E$ and $\approx_L$ are equivalences but not congruences

(not preserved by input prefix and choice)

weak observational congruences:

$p \cong_E q$ iff $\forall \sigma.\ p\sigma \approx_E q\sigma$ and $\forall r.\ p + r \approx_E q + r$

$p \cong_L q$ iff $\forall \sigma.\ p\sigma \approx_L q\sigma$ and $\forall r.\ p + r \approx_L q + r$