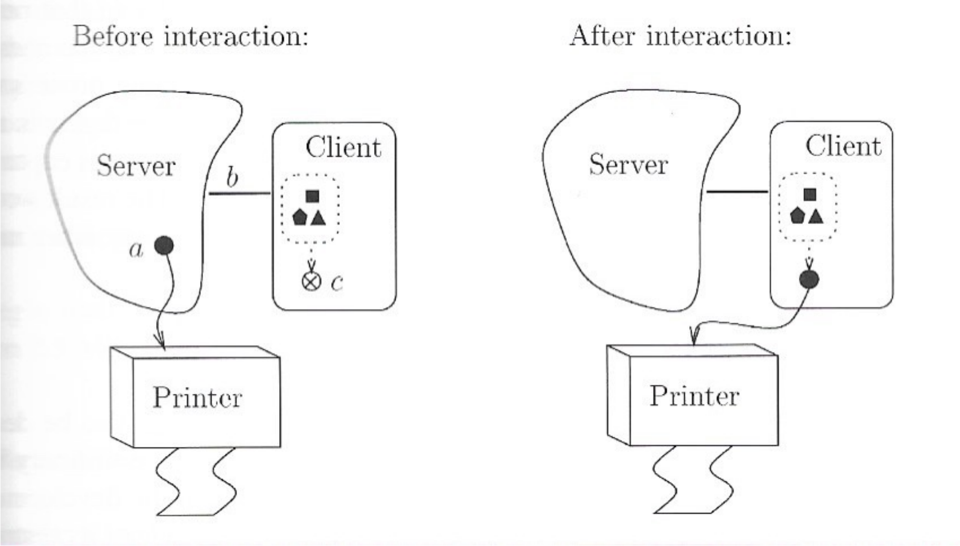
Lezione 29
Cenni al Pi-calcolo

Roberto Gorrieri e Davide Sangiorgi

Da CCS-value passing al pi-calcolo

- Trasmettere valori: ma quali?
- Se l’insieme dei valori trasmittibili è separato dall’insieme dei nomi di canali, allora CCS-VP
  - Struttura del flow graph abbastanza statica (c’è un po’ di dinamicità, ma poco interessante)
- Se si assume che esista un solo insieme di nomi (di canali e/o di valori), allora mobilità di canali!!
- Esempio di server/printer/client

Server-Printer-Client



- $\bar{b}a.S \mid b(c).\bar{c}d.C$ –tau-> $S \mid \bar{a}d.C$

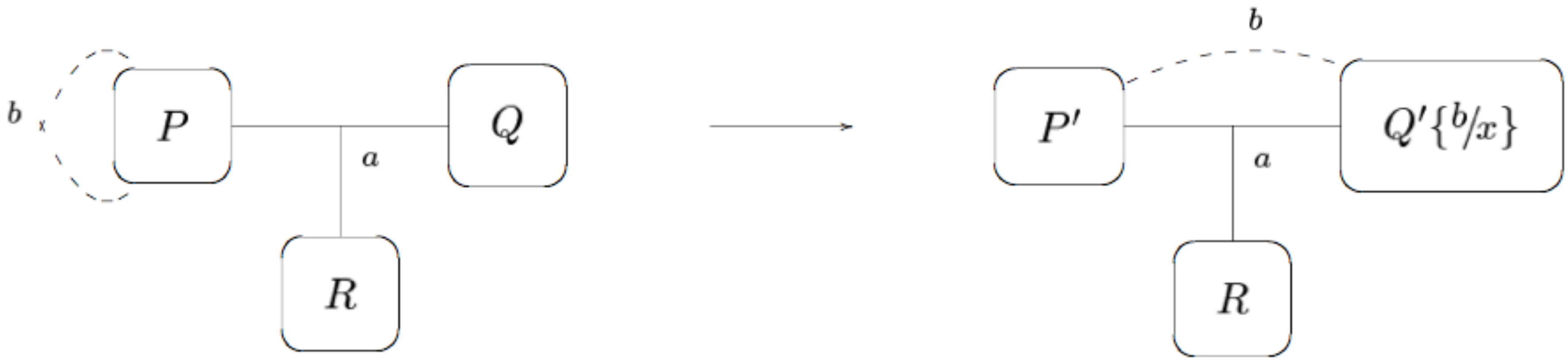
il nome a viene passato dal server al client sul canale comune b

➔ si può instaurare un nuovo canale tra client e printer via a

- $(\nu a)(\bar{b}a.S \mid P) \mid b(c).\bar{c}d.C$ --tau-> $(\nu a)(S|P|\bar{a}d.C)$

se il nome a è inizialmente privato tra server e printer, dopo la comunicazione diventa privato anche col client (**scope extrusion!**)

Mobility IV: movement of *private* links



$$\overbrace{(\boldsymbol{\nu} b)\,(\overline{a}\,b\,.\,P')}^{P} \mid \overbrace{a(x).\,Q'}^{Q} \mid R \longrightarrow (\boldsymbol{\nu} b)\,(P' \mid Q'\{^{b}/_{x}\}) \mid R$$

- Il nome b inizialmente è privato per P
- Spedendo b sul canale a, P estende a Q (ma non a R) lo scope della restrizione

scope extrusion

The π-calculus (core)

$a, b, \ldots, x, y, z, \ldots$			*Names*
			Processes
P	$::=$	0	nil process
	$\mid$	$P \mid P$	parallel
	$\mid$	$(\boldsymbol{\nu} a)\, P$	restriction
	$\mid$	$a(x).\, P$	input
	$\mid$	$\overline{a}\, b\,.\, P$	output
	$\mid$	$!P$	replication ($= P \mid P \ldots$)

– For simplicity (and because less useful), operators of sum and matching are omitted

An attempt at labelled transitions (à la CCS)

inp: $a(b).P \xrightarrow{a(b)} P$ b è bound

out: $\overline{a}\, b\,.\,P \xrightarrow{\overline{a}\, b} P$

parL: $$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

comL: $$\frac{P \xrightarrow{a(x)} P' \qquad Q \xrightarrow{\overline{a}\, b} Q'}{P \mid Q \xrightarrow{\tau} P'\{b/x\} \mid Q'}$$ + simm.

Late semantics

res: $$\frac{P \xrightarrow{\alpha} P'}{\nu a\, P \xrightarrow{\alpha} \nu a\, P'}$$ (α is not an action at a)

a è bound

But then:

- $a(x).P \mid x(y).Q \xrightarrow{a(x)} P \mid x(y).Q$, hence

la sostituzione si applica erroneamente anche a x(y).Q

$$(a(x).P \mid x(y).Q) \mid \overline{a}\, c\,.\,R \xrightarrow{\tau} (P \mid x(y).Q)\{c/x\} \mid R$$

- $\nu b\, \overline{a}\, b\,.\,\overline{b}\, c \xrightarrow{\overline{a}\, b} \nu b\, \overline{b}\, c$, hence

(dynamic binding!!)

$$(\nu b\, \overline{a}\, b\,.\,\overline{b}\, c) \mid a(x).x(y).Q \xrightarrow{\tau} (\nu b\, \overline{b}\, c) \mid b(y).Q$$

Nome bound b estromesso!!

Labeled transition semantics (amended)

Semantica late – regole simmetriche omesse

Abbr.: $\overline{a}(b) \triangleq \nu b\, \overline{a}\, b$

Bound output b

$$\texttt{inp:}\quad a(b).\,P \xrightarrow{a(b)} P \qquad\qquad \texttt{out:}\quad \overline{a}\,b\,.\,P \xrightarrow{\overline{a}\,b} P$$

Bound input b

$$\texttt{parL:}\quad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ \mathrm{bn}(\alpha) \cap \mathrm{fn}(Q) = \emptyset$$

Usa bound output b

$$\texttt{comL:}\quad \frac{P \xrightarrow{a(x)} P' \qquad Q \xrightarrow{\overline{a}\,b} Q'}{P \mid Q \xrightarrow{\tau} P'\{^b/_x\} \mid Q'} \qquad\qquad \texttt{closeL:}\quad \frac{P \xrightarrow{a(x)} P' \qquad Q \xrightarrow{\overline{a}(b)} Q'}{P \mid Q \xrightarrow{\tau} \nu b\,(P'\{^b/_x\} \mid Q')}\ b \notin \mathrm{fn}(P)$$

$$\texttt{res:}\quad \frac{P \xrightarrow{\alpha} P'}{\nu a\, P \xrightarrow{\alpha} \nu a\, P'}\ a \notin \mathrm{n}(\alpha) \qquad\qquad \texttt{open:}\quad \frac{P \xrightarrow{\overline{a}\,b} P'}{\nu b\, P \xrightarrow{\overline{a}(b)} P'}\ a \neq b$$

Genera bound output b

$$\texttt{rep:}\quad \frac{P \mid !P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'} \qquad\qquad \texttt{alpha:}\quad \frac{P =_\alpha Q \qquad Q \xrightarrow{\alpha} Q'}{P \xrightarrow{\alpha} Q'}$$

Esempio1 - alpha-conversione per parL

$$\dfrac{\dfrac{a(x).P \xrightarrow{a(x)} P}{a(x).P \mid Q \xrightarrow{a(x)} P \mid Q}\ \mathbf{x \notin fn(Q)} \qquad \bar{a}u.R \xrightarrow{\bar{a}u} R}{(a(x).P \mid Q) \mid \bar{a}u.R \xrightarrow{tau} (P \mid Q)\{u/x\} \mid R}$$

Ma questo ha senso solo se **$x \notin fn(Q)$**, perché altrimenti la sostituzione $\{u/x\}$ si applicherebbe erroneamente anche a Q. Altrimenti bisogna prima **alpha-convertire** $a(x).P$ **in** $a(z).P\{z/x\}$, **con $z \notin fn(Q)$**

$$\dfrac{\dfrac{\dfrac{a(x).P =_{\alpha} a(z).P\{z/x\} \qquad a(z).P\{z/x\} \xrightarrow{a(z)} P\{z/x\}}{a(x).P \xrightarrow{a(z)} P\{z/x\}}}{a(x).P \mid Q \xrightarrow{a(z)} P\{z/x\} \mid Q}\ \mathbf{z \notin fn(Q)} \qquad \bar{a}u.R \xrightarrow{\bar{a}u} R}{(a(x).P \mid Q) \mid \bar{a}u.R \xrightarrow{tau} (P\{z/x\} \mid Q)\{u/z\} \mid R}$$

Esempio2 - alpha conversione per closeL

Nella regola close-L, se **b $\in$ fn(P)**, è necessario prima alpha-convertire Q =(vb)R con un nuovo bound name c al posto di b, ottenendo (vc)R{c/b}, in modo tale che c $\notin$ fn(P)

$$\frac{(\nu b)R =_{\alpha} (\nu c)R\{c/b\} \qquad (\nu c)R\{c/b\} \xrightarrow{\bar{a}(c)} R'}{(\nu b)R \xrightarrow{\bar{a}(c)} R'}$$ **bound output**

$$\frac{P \xrightarrow{a(x)} P' \qquad (\nu b)R \xrightarrow{\bar{a}(c)} R'}{P \mid (\nu b)R \xrightarrow{tau} (\nu c)(P'\{c/x\} \mid R')} \; \mathbf{c \notin fn(P)}$$

Se c $\in$ fn(P), allora la restrizione su c legherebbe in P' non solo le x sostituite in c, ma erroneamente anche le altre c libere in P'

Esempio3 – altra alpha-conversione per parL (bound output)

$$\dfrac{(\nu b)P \xrightarrow{\bar{a}(b)} P'}{(\nu b)P|Q \xrightarrow{\bar{a}(b)} P'|Q}\ \textcolor{blue}{\mathbf{b \notin fn(Q)}}$$

$$\dfrac{(\nu b)P|Q \xrightarrow{\bar{a}(b)} P'|Q \qquad R \xrightarrow{a(x)} R'}{((\nu b)P|Q)|R \xrightarrow{\tau} (\nu b)((P' \mid Q) \mid R'\{b/x\})}\ \textcolor{red}{\mathbf{b \notin fn(R)}}$$

Ma questo ha senso solo se **$b \notin fn(Q)$**, perché ora Q è sotto restrizione! Altrimenti bisogna prima **alpha-convertire** (vb)P **in** (vc)P{c/b}, **con $c \notin fn(Q)$** (e $c \notin fn(R)$ per la closeR)

(Tau) $\tau.p \xrightarrow{\tau} p$

(In) $x(z).p \xrightarrow{xy} p\{y/z\}$

(Out) $\bar{x}y.p \xrightarrow{\bar{x}y} p$

(Sum) $$\frac{p \xrightarrow{\alpha} p'}{p + q \xrightarrow{\alpha} p'}$$

(Par) $$\frac{p \xrightarrow{\alpha} p'}{p \mid q \xrightarrow{\alpha} p' \mid q} \quad bn(\alpha) \cap fn(q) = \emptyset$$

(Com) $$\frac{p \xrightarrow{\bar{x}y} p' \qquad q \xrightarrow{xy} q'}{p \mid q \xrightarrow{\tau} p' \mid q'}$$

(Close) $$\frac{p \xrightarrow{\bar{x}(w)} p' \qquad q \xrightarrow{xw} q'}{p \mid q \xrightarrow{\tau} (\nu w)(p' \mid q')} \quad w \notin fn(q)$$

(Res) $$\frac{p \xrightarrow{\alpha} p'}{(\nu y)p \xrightarrow{\alpha} (\nu y)p'} \quad y \notin n(\alpha)$$

(Open) $$\frac{p \xrightarrow{\bar{x}y} p'}{(\nu y)p \xrightarrow{\bar{x}(w)} p'\{w/y\}} \quad y \neq x \wedge w \notin fn((\nu y)p')$$

Semantica op. early

- No alfa-conversione (implementata implicitamente per mezzo di (In) e (Open))
- Input label aggiuntiva (**not bound**!)
- x(z).p –xy-> p{y/z} è molto diversa da x(z).p -x(z)-> p
- Regola (Com) à la CCS
- Regole simmetriche omesse
- Regola per !p omessa

Semantica late alternativa

(senza alfa-conversione e con regole alternative per replicazione)

$$\text{(OUTPUT-ACT)}\ \frac{}{\bar{x}z.P \xrightarrow{\bar{x}z} P} \qquad \text{(INPUT-ACT)}\ \frac{w \notin fn((y)P)}{x(y).P \xrightarrow{x(w)} P\{w/y\}}$$

$$\text{(PAR)}\ \frac{P \xrightarrow{\alpha} P' \qquad bn(a) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \text{(COM)}\ \frac{P \xrightarrow{\bar{x}z} P' \qquad Q \xrightarrow{x(y)} Q'}{P|Q \xrightarrow{\tau} P'|Q'\{z/y\}}$$

$$\text{(CLOSE)}\ \frac{P \xrightarrow{\bar{x}(w)} P' \qquad Q \xrightarrow{x(w)} Q'}{P|Q \xrightarrow{\tau} (w)(P'|Q')} \qquad \text{(RES)}\ \frac{P \xrightarrow{\alpha} P' \qquad y \notin n(a)}{(y)P \xrightarrow{\alpha} (y)P'}$$

$$\text{(OPEN)}\ \frac{P \xrightarrow{\bar{x}y} P' \qquad y \neq x \qquad w \notin fn((y)P')}{(y)P \xrightarrow{\bar{x}(w)} P'\{w/y\}} \qquad \text{(REP-ACT)}\ \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'|!P}$$

$$\text{(REP-COMM)}\ \frac{P \xrightarrow{\bar{x}z} P' \qquad P \xrightarrow{x(y)} P''}{!P \xrightarrow{\tau} (P'|P''\{z/y\})|!P} \qquad \text{(REP-CLOSE)}\ \frac{P \xrightarrow{\bar{x}(w)} P' \qquad P \xrightarrow{x(w)} P''}{!P \xrightarrow{\tau} ((w)(P'|P''))|!P}$$

Table 1. SOS rules for the synchronous mini-π-calculus. PAR, COM and CLOSE also have symmetric rules.

Reduction semantics

[Milner '91, following: Gamma language, Banâtre & Le Métayer '88, Chemical Abstract Machine, Berry & Boudol '90]

Structural congruence The smallest congruence $\equiv$ s.t.:

1. $P \equiv Q$ if P and Q are alpha convertible

2. $P \mid 0 \equiv P$, $P \mid Q \equiv Q \mid P$, $P \mid (Q \mid R) \equiv (P \mid Q) \mid R$

3. $\boldsymbol{\nu} x\, 0 \equiv 0$, $\boldsymbol{\nu} x\, \boldsymbol{\nu} y\, P \equiv \boldsymbol{\nu} y\, \boldsymbol{\nu} x\, P$

4. $\boldsymbol{\nu} x\, (P \mid Q) \equiv (\boldsymbol{\nu} x\, P) \mid Q$, if x not free in Q

5. $!P \equiv P \mid !P$

Reduction relation ($\longrightarrow$)

$$\text{R-INTER} \quad \frac{}{(\overline{x}\, y \,.\, P_1) \mid (x(z).\, P_2) \longrightarrow P_1 \mid P_2\{y/z\}}$$

$$\text{R-PAR} \quad \frac{P_1 \longrightarrow P_1'}{P_1 \mid P_2 \longrightarrow P_1' \mid P_2} \qquad\qquad \text{R-RES} \quad \frac{P \longrightarrow P'}{\boldsymbol{\nu} z\, P \longrightarrow \boldsymbol{\nu} z\, P'}$$

$$\text{R-STRUCT} \quad \frac{P_1 \equiv P_2 \longrightarrow P_2' \equiv P_1'}{P_1 \longrightarrow P_1'}$$

Teorema: P→P’ iff P-tau->P’’ congruente a P’

Reduction semantics: examples

$$
\begin{array}{rcl}
x(y).\,P \mid R \mid \overline{x}\,z\,.\,Q & \equiv & \\
\overline{x}\,z\,.\,Q \mid x(y).\,P \mid R & \longrightarrow & \\
Q \mid P\{^z/_x\} \mid R & \equiv & P \\
\{^z/_x\} \mid R \mid Q & &
\end{array}
$$

Let $P \triangleq \boldsymbol{\nu} y\,(\overline{x}\,y\,.\,0)$.

$$
\begin{array}{rc}
x(z).\,\overline{a}\,z\,.\,0 \mid !P & \equiv \\
x(z).\,\overline{a}\,z\,.\,0 \mid \boldsymbol{\nu} y\,(\overline{x}\,y\,.\,0) \mid !P & \equiv \\
\boldsymbol{\nu} y\,(x(z).\,\overline{a}\,z\,.\,0 \mid \overline{x}\,y\,.\,0) \mid !P & \equiv \\
\boldsymbol{\nu} y\,(\overline{x}\,y\,.\,0 \mid x(z).\,\overline{a}\,z\,.\,0) \mid !P & \longrightarrow \\
\boldsymbol{\nu} y\,(0 \mid \overline{a}\,y\,.\,0) \mid !P & \equiv \\
\boldsymbol{\nu} y\,(\overline{a}\,y\,.\,0) \mid !P &
\end{array}
$$

Weak barbed bisimulation

- A process P has a **strong** barb on x ∈ N, denoted by P↓x, iff there is a P′ with P−x(y)→P′ for some y∈N.
- It has a **strong** barb on x̄ ,P↓x̄ , iff there is a P′ with P−x̄ z →P′ or P−x̄ (z) →P′ for some z∈N.
- A process P has a **weak** barb on a (a ∈ {x, x̄ | x ∈ N}), P⇓a, iff there is a P′ such that P −τ*→ P′ and P′↓a.

Definition 8. A symmetric relation $\mathcal{R}$ on $\mathcal{P}_\pi$ is a *weak barbed bisimulation* iff $P \mathcal{R} Q$ implies

1. if $P{\downarrow_a}$ with $a \in \{x, \bar{x} \mid x \in \mathcal{N}\}$ then $Q{\Downarrow_a}$ and
2. if $P \xrightarrow{\tau} P'$ then a Q' exists with $Q \xrightarrow{\tau}^{*} Q'$ and $P' \mathcal{R} Q'$.

The largest weak barbed bisimulation is denoted by $\stackrel{\bullet}{\approx}$, or $\approx_{\mathrm{WBB}}$.

Weak Barbed congruence

- Nota che l'equivalenza ha poco senso:
- $\bar{x}z.0$ e $\bar{x}y.0$ e $(\nu z)\,\bar{x}z.0$ sono tutti barbed equivalenti
- Tuttavia, la congruenza indotta ha molto senso: la (weak) barbed congruence è la congruenza ottenuta chiudendo per contesti la (weak) barbed bisimulation equivalence.
- Coincide con l'equivalenza (weak) **early** bisimulation congruence sul labeled transition system del pi-calcolo.

Problemi per **labeled** semantics(1)

- $P = a(b).0$ e $Q=a(x).(\nu c).\bar{c}b$

P e Q intuitivamente sono equivalenti: fanno un input e poi terminano.

Tuttavia, P—a(b)-> 0 mentre Q non può, nemmeno con alpha-conversione, perché $b \in fn(Q)$ (Esercizio: Fare esempio in cui $b \in bn(Q)$)

Questa differenza non è importante: se P può fare a(b), allora, per alpha-conversione, può fare a(w) per infiniti nomi w!

A Q basta simulare **ogni azione bound** (bound input e bound output per late sem.op., solo bound output per early sem.op.) dove l'oggetto bound non sia usato in Q (cioè né libero né bound).

Sotto questa ipotesi, P e Q sono equivalenti perché Q non necessita di match-are la transizione a(b) di P.

Problemi per **labeled** semantics(2)

- $P = (\nu b)\bar{c}b.0$ e $Q = (\nu a)(\bar{c}a.0 \mid (\nu b).\bar{b}a)$

P e Q intuitivamente sono equivalenti: fanno un bound output sul canale c e poi terminano.

Tuttavia, $P \xrightarrow{\bar{c}(b)} (\nu b)0$ mentre Q non può, nemmeno con alpha-conversione, perché $b \in bn(Q)$ (Esercizio: Fare esempio in cui $b \in fn(Q)$)

Questa differenza non è importante: se P può fare $\bar{c}(b)$, allora, per alpha-conversione, può fare $\bar{c}(w)$ per infiniti nomi w!

A Q basta simulare **ogni azione bound** (bound input e bound output per late op., solo bound output per early op.) dove l'oggetto bound non sia usato in Q.

Sotto questa ipotesi, P e Q sono equivalenti perché Q non necessita di match-are la transizione $\bar{c}(b)$ di P.

Problemi per labeled **late** semantics – come gestire il bound input?

- P –a(x)-> P'   come risponde Q?
- In caso strong, Q—a(x)->Q'

Però il nome x è un place-holder per qualsiasi cosa possa essere ricevuta.

Allora il comportamento di P' deve essere considerato rispetto a tutte le sostituzioni {u/x} e dobbiamo richiedere che (*due alternative!*):

- Early: per ogni sostituzione {u/x}, esista un Q' tale che Q—a(x)->Q' e (P'{u/x},Q' {u/x}) in R
- Late: esista un Q' tale che Q—a(x)->Q' e per ogni sostituzione {u/x}, (P'{u/x},Q' {u/x}) in R

Early weak bisimulation
(su sem.op. late)

Definition 7. A symmetric binary relation $\mathcal{R}$ on π-processes P, Q is a *early weak bisimulation* iff $P \mathrel{\mathcal{R}} Q$ implies

1. if $P \xrightarrow{\tau} P'$ then a Q' exists with $Q \xrightarrow{\tau}{}^{*} Q'$ and $P' \mathrel{\mathcal{R}} Q'$,
2. if $P \xrightarrow{\alpha} P'$ where $\alpha = \bar{x}z$ or $\bar{x}(y)$ with $y \notin n(P) \cup n(Q)$, then a Q' exists with $Q \xrightarrow{\tau}{}^{*} \xrightarrow{\alpha} \xrightarrow{\tau}{}^{*} Q'$ and $P' \mathrel{\mathcal{R}} Q'$,
3. if $P \xrightarrow{x(y)} P'$ with $y \notin n(P) \cup n(Q)$ then for all w a Q' exists satisfying $Q \xrightarrow{\tau}{}^{*} \xrightarrow{x(y)} \xrightarrow{\tau}{}^{*} Q'$ and $P'\{w/y\} \mathrel{\mathcal{R}} Q'\{w/y\}$.

We denote the largest early weak bisimulation by $\approx_{\mathrm{EWB}}$.

- Early (**strong**) bisimulation è una equivalenza ed anche una congruenza per tutti gli operatori tranne il prefisso di input.

Prefisso di input: no congruence

- $a \mid \bar{b}$ è strong bisimile a $a.\bar{b} + \bar{b}.a$
- Tuttavia

$c(a).(a \mid \bar{b})$ non è strong bisimile a $c(a).(a.\bar{b} + \bar{b}.a)$ perché il primo può fare $c(b)$ raggiungendo lo stato $(b \mid \bar{b})$, mentre il secondo, facendo $c(b)$, diventa $(b.\bar{b} + \bar{b}.b)$, che non è strong bisimile a $(b \mid \bar{b})$

Per ottenere la congruenza, bisogna "chiudere" l'equivalenza rispetto alle sostituzioni:

P è **congruente** a Q sse $P\sigma$ è equivalente a $Q\sigma$ per ogni sostituzione σ.

(Early **weak** bisimulation non è una congruenza anche per il +, come al solito.)

Tante semantiche ….

- Abbiamo visto la weak early bisimilarity su sem.op di tipo late. Ma è possibile definire anche
- (weak) late bisimilarity su sem.op. di tipo late,

Le semantiche behavioral di tipo late sono leggermente più fini di quelle early.

- Ed altre ancora …. (e.g., **open** di Sangiorgi che richiede di effettuare la chiusura per sostituzioni già nella definizione di bisimulazione).
- Una particolarmente semplice è bisimilarity su sem.op. di tipo early (che coincide con early bisimilarity su sem.op. late)

(Tau) $\tau.p \xrightarrow{\tau} p$

(In) $x(z).p \xrightarrow{xy} p\{y/z\}$

(Out) $\bar{x}y.p \xrightarrow{\bar{x}y} p$

(Sum) $\dfrac{p \xrightarrow{\alpha} p'}{p+q \xrightarrow{\alpha} p'}$

(Par) $\dfrac{p \xrightarrow{\alpha} p'}{p\,|\,q \xrightarrow{\alpha} p'\,|\,q}$ $bn(\alpha) \cap fn(q) = \emptyset$

(Com) $\dfrac{p \xrightarrow{\bar{x}y} p' \qquad q \xrightarrow{xy} q'}{p\,|\,q \xrightarrow{\tau} p'\,|\,q'}$

(Close) $\dfrac{p \xrightarrow{\bar{x}(w)} p' \qquad q \xrightarrow{xw} q'}{p\,|\,q \xrightarrow{\tau} (\nu w)(p'\,|\,q')}$ $w \notin fn(q)$

(Res) $\dfrac{p \xrightarrow{\alpha} p'}{(\nu y)p \xrightarrow{\alpha} (\nu y)p'}$ $y \notin n(\alpha)$

(Open) $\dfrac{p \xrightarrow{\bar{x}y} p'}{(\nu y)p \xrightarrow{\bar{x}(w)} p'\{w/y\}}$ $y \neq x \wedge w \notin fn((\nu y)p')$

Esempio di behavioral semantics su sem.op. early:

strong early bisimulation

(caso molto semplice)

n(p,q) = fn(p) ∪ fn(q) ∪ bn(q)

Definition 2.1 A binary relation R over the set of terms is an *early bisimulation* if $(p, q) \in R$ implies:

- if $p \xrightarrow{\alpha} p'$ and $bn(\alpha) \cap$ n(p,q) $= \emptyset$ there exists then q' such that $q \xrightarrow{\alpha} q'$ and $(p', q') \in R$;
- symmetrically for q derivations.

Two terms p and q are *bisimilar*, written $p \sim q$, if there exists a bisimulation R such that $(p, q) \in R$. $\square$