Universita' di Bologna

Facolta' di scienze matematiche fisiche e naturali

Corso di laurea magistrale in Scienze Informatiche

Tesi di laurea

Multi π calcolo

Candidato:
Federico Viscomi

Tutore
Prof. Roberto Gorrieri

. . . .

./pi.jpg

Anno accademico 20011/2012

0.1 Abstract

Il π calcolo e' un formalismo che descrive e analizza le proprieta' del calcolo concorrente. Nasce come proseguio del lavoro gia' svolto sul CCS (Calculus of Communicating Systems). L'aspetto appetibile del π calcolo rispetto ai formalismi precedenti e' l'essere in grado di descrivere la computazione concorrente in sistemi la cui configurazione puo' cambiare nel tempo. Nel CCS e nel π calcolo manca la possibilta' di modellare sequenze atomiche di azioni e di modellare la sincronizzazione multiparte. Il Multi CCS [3] estende il CCS con un'operatore di strong prefixing proprio per colmare tale vuoto. In questa tesi si cerca di trasportare per analogia le soluzioni introdotte dal Multi CCS verso il π calcolo. Il risultato finale e' un linguaggio chiamato Multi π calcolo.

In particolare il Multi π calcolo permette la sincronizzazione transazionale e la sincronizzazione multiparte. aggiungere una sintesi brevissima dei risultati ottenuti sul Multi π calcolo.

Contents

Chapter 1

TODO

1. dimostrare(o negare) l'equivalenza del pi calcolo con e senza congruenza strutturale e con e senza alfa conversione. FATTO MA NON COME SPERATO.

2. nel multi pi calcolo con strong prefixing solo su input o solo su output: definire una semantica di basso livello sulla falsariga di quell'articolo. FATTO MA NON COME SPERATO. raggiungere un qualche risultato simile anche per multipiInpOut

3. terminare la parte sulle bisimulazioni nel multipiOut senza congruenza strutturale. fare una cosa simile anche per multipiInp senza congruenza strutturale?

4. terminare la parte sulle bisimulazioni nel multipiInp con congruenza strutturale. fare una cosa simile anche per multipiOut con congruenza strutturale?

5. dare una semantica open step e provare a definire una bisimulazione open sulla semantica step. per multipiOut con e senza congruenza strutturale e per multipiInp con e senza congruenza strutturale

6. trovare la congruenza coarsest contenuta nella bisimulazione scelta in precedenza

7. ripetere i ragionamenti fatti in precedenza anche per multipiInpOut

8. ha senso una semantica che conserva la proprieta' di essere una forma normale definita su coppie (insieme di nomi ristretti, processo senza restrizioni unguarded)?

Chapter 2

Π calculus

The π calculus is a mathematical model of processes whose interconnections change as they interact. The basic computational step is the transfer of a communication link between two processes. The idea that the names of the links belong to the same category as the transferred objects is one of the cornerstone of the calculus. The π calculus allows channel names to be communicated along the channels themselves, and in this way it is able to describe concurrent computations whose network configuration may change during the computation.

A coverage of π calculus is on [4], [5] and [7]

2.1 Syntax

We suppose that we have a countable set of names $\mathbb{N}$, ranged over by lower case letters $a, b, \cdots, z$. This names are used for communication channels and values. Furthermore we have a set of identifiers, ranged over by A. We represent the agents or processes by upper case letters $P, Q, \cdots$. A process can perform the following actions:

$$\pi ::= \overline{x}y \mid x(z) \mid \tau$$

The process are defined by the following grammar:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P + Q \mid (\nu x)P \mid A(\tilde{x})$$

and they have the following intuitive meaning:

0 is the empty process which cannot perform any actions

$\pi.P$ is an action prefixing, this process can perform action π e then behave like P, the action can be:

- $\overline{x}y$ is an output action, this sends the name y along the name x. We can think about x as a channel or a port, and about y as an output datum sent over the channel
- $x(z)$ is an input action, this receives a name along the name x. z is a variable which stores the received data.
- τ is a silent or invisible action, this means that a process can evolve to P without interaction with the environment

for any action which is not a τ, the first name that appears in the action is called subject of the action and the second name is called object of the action.

$P + Q$ is the sum, this process can enact either P or Q

$P|Q$ is the parallel composition, P and Q can execute concurrently and also synchronize with each other

$$
\begin{array}{llll}
B(0,I) & = \emptyset & B(Q+R,I) & = B(Q,I)\cup B(R,I) \\
B(\overline{x}y.Q,I) & = B(Q,I) & B(Q|R,I) & = B(Q,I)\cup B(R,I) \\
B(x(y).Q,I) & = \{y,\overline{y}\}\cup B(Q,I) & B((\nu x)Q,I) & = \{x,\overline{x}\}\cup B(Q,I) \\
B(\tau.Q,I) & = B(Q,I) & &
\end{array}
$$

$$
B(A,I) = \begin{cases} B(Q, I\cup\{A\})\ where\ A(\tilde{x}) \stackrel{def}{=} Q & if\ A\notin I \\ \emptyset & if\ A\in I \end{cases}
$$

Table 2.1: Bound occurrences

$$
\begin{array}{lll}
fn(\overline{x}y.Q) = \{x,\overline{x},y,\overline{y}\}\cup fn(Q) & fn(Q+R) = fn(Q)\cup fn(R) & fn(0) = \emptyset \\
fn(x(y).Q) = \{x,\overline{x}\}\cup (fn(Q)-\{y,\overline{y}\}) & fn(Q|R) = fn(Q)\cup fn(R) & \\
fn((\nu x)Q) = fn(Q)-\{x,\overline{x}\} & fn(\tau.Q) = fn(Q) & \dfrac{A(\tilde{x}) \stackrel{def}{=} P}{fn(A)=\{\tilde{x}\}}
\end{array}
$$

Table 2.2: Free occurrences

$(\nu z)P$ is the scope restriction. This process behave as P but the name z is local. This process cannot use the name z to interact with other processes.

$A(\tilde{x})$ is an identifier. Every identifier has a definition

$$A(x_1,\cdots,x_n) = P$$

the x_is must be pairwise different. The intuition is that we can substitute for some of the x_is in P to get a π calculus process. We can write $\tilde{x}$ for $x_1,\cdots,x_n$.

To resolve ambiguity we can use parenthesis and observe the conventions that prefixing and restriction bind more tightly than composition and prefixing binds more tightly than sum.

Definition 2.1.1. We say that the input prefix $x(z).P$ *binds* z in P or is a *binder* for z in P. We also say that P is the *scope* of the binder and that any occurrence of z in P are *bound* by the binder. Also the restriction operator $(\nu z)P$ is a binder for z in P.

Definition 2.1.2. $bn(P)$ is the set of names that have a bound occurrence in P and is defined as $B(P,\emptyset)$, where $B(P,I)$, with I a set of identifiers, is defined in table 2.1

Definition 2.1.3. We say that a name x is *free* in P if P contains a non bound occurrence of x. We write $fn(P)$ for the set of names with a free occurrence in P. $fn(P)$ is defined in table 2.2

Definition 2.1.4. $n(P)$ which is the set of all names in P and is defined in the following way:

$$n(P) = fn(P)\cup bn(P)$$

Definition 2.1.5. We say that τ and actions which does not have any binder, such as $xy, \overline{x}y$, are *free* actions. Whether the other actions are *bound* actions.

In a definition $A(\tilde{x}) = P$ the $\tilde{x}$ are exactly the free names contained in P, specifically $fn(P) = \{\tilde{x}\}$. If we look at the definitions of *bn* and of fn we notice that if P contains another identifier whose definition is: $B(\tilde{z}) = Q$ then we have $fn(Q) \subseteq \{\tilde{x}\}$.

$$0\{b/a\} \;=\; 0 \qquad (\overline{x}y.Q)\{b/a\} \;=\; \overline{x}\{b/a\}y\{b/a\}.Q\{b/a\} \qquad (\tau.Q)\{b/a\} \;=\; \tau.Q\{b/a\}$$

$$\frac{y \neq a \qquad y \neq b}{(x(y).Q)\{b/a\} \;=\; x\{b/a\}(y).Q\{b/a\}} \qquad \frac{c \notin n(x(b).Q)}{(x(b).Q)\{b/a\} \;=\; x\{b/a\}(c).((Q\{c/b\})\{b/a\})}$$

$$(x(a).Q)\{b/a\} \;=\; x\{b/a\}(a).Q$$

$$\frac{a \in \tilde{x} \qquad A(\tilde{x}) \stackrel{def}{=} P}{A(\tilde{x})\{b/a\} = A(\tilde{x}\{b/a\})}$$

$$(Q + R)\{b/a\} \;=\; Q\{b/a\} + R\{b/a\} \qquad (Q|R)\{b/a\} \;=\; Q\{b/a\}|R\{b/a\}$$

$$\frac{y \neq a \qquad y \neq b}{((\nu y)Q)\{b/a\} \;=\; (\nu y)Q\{b/a\}} \qquad ((\nu a)Q)\{b/a\} \;=\; (\nu a)Q$$

$$\frac{c \notin n((\nu b)Q) \qquad a \in fn(Q)}{((\nu b)Q)\{b/a\} \;=\; (\nu c)((Q\{c/b\})\{b/a\})}$$

Table 2.3: Syntatic substitution

Definition 2.1.6. $P\{b/a\}$ is the syntactic substitution of name b for a different name a inside a π calculus process and it consists in replacing every free occurrences of a with b. If b is a bound name in P, in order to avoid name capture we perform an appropriate α conversion. $P\{b/a\}$ is defined in table 2.3. We use the notation $\{\tilde{x}/\tilde{y}\}$ as a short for $\{x_1/y_1, \cdots, x_n/y_n\}$ which is not the composition of the substitutions $\{x_1/y_1\} \circ \ldots \circ \{x_n/y_n\}$

2.2 Operational Semantic(without structural congruence)

2.2.1 Early operational semantic(without structural congruence)

The semantic of a π calculus process is a labeled transition system such that:

- the nodes are π calculus process. The set of node is $\mathbb{P}$
- the actions can be:
  - unbound input xy
  - unbound output $\overline{x}y$
  - the silent action τ
  - bound output $\overline{x}(y)$

  The set of actions is $\mathbb{A}$, we use α to range over the set of actions.
- the transition relations is $\rightarrow \subseteq \mathbb{P} \times \mathbb{A} \times \mathbb{P}$

Definition 2.2.1. The *early transition relation* $\rightarrow \subseteq \mathbb{P} \times \mathbb{A} \times \mathbb{P}$ is the smallest relation induced by the rules in table 2.4.

Example We show now an example of the so called scope extrusion, in particular we prove that

$$a(x).P \mid (\nu b)\overline{a}b.Q \xrightarrow{\tau} (\nu b)(P\{b/x\} \mid Q)$$

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{SumL}\ \frac{P \xrightarrow{\alpha} P^{'}}{P+Q \xrightarrow{\alpha} P^{'}} \qquad \textbf{SumR}\ \frac{Q \xrightarrow{\alpha} Q^{'}}{P+Q \xrightarrow{\alpha} P^{'}}$$

$$\textbf{ParL}\ \frac{P \xrightarrow{\alpha} P^{'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P^{'}|Q} \qquad \textbf{ParR}\ \frac{Q \xrightarrow{\alpha} Q^{'} \qquad bn(\alpha) \cap fn(P) = \emptyset}{P|Q \xrightarrow{\alpha} P|Q^{'}}$$

$$\textbf{Res}\ \frac{P \xrightarrow{\alpha} P^{'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P^{'}}$$

$$\textbf{ResAlp1}\ \frac{(\nu w)P\{w/z\} \xrightarrow{\alpha} P^{'} \qquad w \notin n(P)}{(\nu z)P \xrightarrow{\alpha} P^{'}} \qquad \textbf{ResAlp2}\ \frac{P \xrightarrow{\alpha} P^{'} \qquad w \notin n(P)}{(\nu w)P\{w/z\} \xrightarrow{\alpha} (\nu w)P^{'}}$$

$$\textbf{EComL}\ \frac{P \xrightarrow{xy} P^{'} \qquad Q \xrightarrow{\overline{x}y} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}|Q^{'}} \qquad \textbf{EComR}\ \frac{P \xrightarrow{\overline{x}y} P^{'} \qquad Q \xrightarrow{xy} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}|Q^{'}}$$

$$\textbf{ClsL}\ \frac{P \xrightarrow{\overline{x}(z)} P^{'} \qquad Q \xrightarrow{xz} Q^{'} \qquad z \notin fn(Q)}{P|Q \xrightarrow{\tau} (\nu z)(P^{'}|Q^{'})} \qquad \textbf{ClsR}\ \frac{P \xrightarrow{xz} P^{'} \qquad Q \xrightarrow{\overline{x}(z)} Q^{'} \qquad z \notin fn(P)}{P|Q \xrightarrow{\tau} (\nu z)(P^{'}|Q^{'})}$$

$$\textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P^{'} \qquad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P^{'}} \qquad \textbf{OpnAlp}\ \frac{(\nu w)P\{w/z\} \xrightarrow{\overline{x}(w)} P^{'} \qquad w \notin n(P) \qquad x \neq w \neq z}{(\nu z)P \xrightarrow{\overline{x}(w)} P^{'}}$$

$$\textbf{Ide}\ \frac{A(\tilde{x}) \overset{def}{=} P \qquad P\{\tilde{w}/\tilde{x}\} \xrightarrow{\alpha} P^{'}}{A(\tilde{x})\{\tilde{w}/\tilde{x}\} \xrightarrow{\alpha} P^{'}}$$

Table 2.4: Early semantic without structural congruence and without explicit α conversion

where we suppose that $b \notin fn(P)$. In this example the scope of (νb) moves from the right hand component to the left hand.

$$\textsc{CloseR}\ \dfrac{\textsc{EInp}\ \dfrac{}{a(x).P \xrightarrow{ab} P\{b/x\}} \qquad \textsc{Opn}\ \dfrac{\textsc{Out}\ \dfrac{}{\overline{a}b.Q \xrightarrow{\overline{a}b} Q} \qquad a \neq b}{(\nu b)\overline{a}b.Q \xrightarrow{\overline{a}(b)} Q} \qquad b \notin fn((\nu b)\overline{a}b.Q)}{a(x).P \mid (\nu b)\overline{a}b.Q \xrightarrow{\tau} (\nu b)(P\{b/x\} \mid Q)}$$

Example We want to prove now that:

$$((\nu b)a(x).P) \mid \overline{a}b.Q \xrightarrow{\tau} ((\nu c)(P\{c/b\}\{b/x\}))|Q$$

where $b \notin bn(P)$

$$\textsc{ResAlp1}\ \dfrac{\textsc{Res}\ \dfrac{\textsc{EInp}\ \dfrac{}{(a(x).P)\{c/b\} \xrightarrow{ab} P\{c/b\}\{b/x\}} \qquad c \notin n(a(b))}{(\nu c)((a(x).P)\{c/b\}) \xrightarrow{ab} (\nu c)(P\{c/b\}\{b/x\})} \qquad b \notin n((a(x).P)\{c/b\})}{(\nu b)a(x).P \xrightarrow{ab} (\nu c)P\{c/b\}\{b/x\}}$$

$$\textsc{EComL}\ \dfrac{(\nu b)a(x).P \xrightarrow{ab} (\nu c)P\{c/b\}\{b/x\} \qquad \textsc{EOut}\ \dfrac{}{\overline{a}b.Q \xrightarrow{\overline{a}b} Q}}{((\nu b)a(x).P) \mid \overline{a}b.Q \xrightarrow{\tau} ((\nu c)(P\{c/b\}\{b/x\}))|Q}$$

Example We have to spend some time to deal with the change of bound names in an identifier. Suppose we have

$$A(x) \stackrel{def}{=} \underbrace{x(y).x(a).0}_{P}$$

From the definition of substitution it follows that $A(x)\{y/x\} = A(y)$. The identifier $A(y)$ is expected to behave consistently with $P\{y/x\} = y(z).y(a).0$. So for example we have to prove that $A(y) \xrightarrow{yw} y(a).0$. We can prove this in the following way:

$$\textbf{Ide}\ \dfrac{A(x) \stackrel{def}{=} P \qquad \textbf{EInp}\ \dfrac{}{P\{y/x\} \xrightarrow{yw} y(a).0}}{A(y) \xrightarrow{yw} y(a).0}$$

2.2.2 Late operational semantic(without structural congruence)

In this case the set of actions $\mathbb{A}$ contains

- bound input $x(y)$
- unbound output $\overline{x}y$
- the silent action τ
- bound output $\overline{x}(y)$

Definition 2.2.2. The *late transition relation without structural congruence* $\rightarrow \subseteq \mathbb{P} \times \mathbb{A} \times \mathbb{P}$ is the smallest relation induced by the rules in table 2.5.

2.3 Structural congruence

Structural congruences are a set of equations defining equality and congruence relations on process. They can be used in combination with an SOS semantic for languages. In some cases structural congruences help simplifying the SOS rules: for example they can capture inherent properties of composition operators(e.g. commutativity, associativity and zero element). Also, in process calculi,

$$\textbf{LInp}\ \frac{z \notin fn(P)}{x(y).P \xrightarrow{x(z)} P\{z/y\}} \qquad \textbf{Res}\ \frac{P \xrightarrow{\alpha} P' \quad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P'}$$

$$\textbf{SumL}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \textbf{SumR}\ \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

$$\textbf{ParL}\ \frac{P \xrightarrow{\alpha} P' \quad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \textbf{ParR}\ \frac{Q \xrightarrow{\alpha} Q' \quad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P|Q'}$$

$$\textbf{ComL}\ \frac{P \xrightarrow{x(y)} P' \quad Q \xrightarrow{\overline{x}(z)} Q'}{P|Q \xrightarrow{\tau} P'\{z/y\}|Q'} \qquad \textbf{ComR}\ \frac{P \xrightarrow{\overline{x}(z)} P' \quad Q \xrightarrow{x(y)} Q'}{P|Q \xrightarrow{\tau} P'|Q'\{z/y\}}$$

$$\textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P' \quad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P'} \qquad \textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P}$$

$$\textbf{ClsL}\ \frac{P \xrightarrow{\overline{x}(z)} P' \quad Q \xrightarrow{xz} Q' \quad z \notin fn(Q)}{P|Q \xrightarrow{\tau} (\nu z)(P'|Q')} \qquad \textbf{ClsR}\ \frac{P \xrightarrow{xz} P' \quad Q \xrightarrow{\overline{x}(z)} Q' \quad z \notin fn(P)}{P|Q \xrightarrow{\tau} (\nu z)(P'|Q')}$$

$$\textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textbf{Ide}\ \frac{A(\tilde{x}) \overset{def}{=} P \quad P\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha} P'}{A(\tilde{y}) \xrightarrow{\alpha} P'}$$

Table 2.5: Late semantic without structural congruence and without explicit α conversion

structural congruences let processes interact even in case they are not adjacent in the syntax. There is a possible trade off between what to include in the structural congruence and what to include in the transition rules: for example in the case of the commutativity of the sum operator. It is worth noticing that in most process calculi every structurally congruent processes should never be distinguished and thus any semantic must assign them the same behaviour.

Definition 2.3.1. A *change of bound names* in a process P is the replacement of a subterm $x(z).Q$ of P by $x(w).Q\{w/z\}$ or the replacement of a subterm $(\nu z)Q$ of P by $(\nu w)Q\{w/z\}$ where in each case w does not occur in Q.

Definition 2.3.2. A *context* $C[\cdot]$ is a process with a placeholder. If $C[\cdot]$ is a context and we replace the placeholder with P, than we obtain $C[P]$. In doing so, we make no α conversions.

Definition 2.3.3. A *congruence* is a binary relation on processes such that:

- S is an equivalence relation
- S is preserved by substitution in contexts: for each pair of processes (P, Q) and for each context $C[\cdot]$

$$(P, Q) \in S \ \Rightarrow\ (C[P], C[Q]) \in S$$

Definition 2.3.4. Processes P and Q are α *convertible* or α *equivalent* if Q can be obtained from P by a finite number of changes of bound names. If P and Q are α equivalent then we write $P \equiv_\alpha Q$. Specifically the α equivalence is the smallest binary relation on processes that satisfies the laws in table 2.6. In a process P we can assume that all bound names are different.

It remains the problem of proving that α equivalence is well defined, i.e. if we change only some bound names in a process P then we get a process α equivalent to P.

According to [2] the following lemma holds:

$$\textsc{AlpOut}\ \frac{P \equiv_\alpha Q}{\overline{x}y.P \equiv_\alpha \overline{x}y.Q} \qquad \textsc{AlpTau}\ \frac{P \equiv_\alpha Q}{\tau.P \equiv_\alpha \tau.Q} \qquad \textsc{AlpInp}\ \frac{P \equiv_\alpha Q}{x(y).P \equiv_\alpha x(y).Q}$$

$$\textsc{AlpIde}\ \frac{}{A(\tilde{x}) \equiv_\alpha A(\tilde{x})} \qquad \textsc{AlpZero}\ \frac{}{0 \equiv_\alpha 0} \qquad \textsc{AlpRes}\ \frac{P \equiv_\alpha Q}{(\nu x)P \equiv_\alpha (\nu x)Q}$$

$$\textsc{AlpPar}\ \frac{P_1 \equiv_\alpha Q_1 \qquad P_2 \equiv_\alpha Q_2}{P_1|P_2 \equiv_\alpha Q_1|Q_2} \qquad \textsc{AlpSum}\ \frac{P_1 \equiv_\alpha Q_1 \qquad P_2 \equiv_\alpha Q_2}{P_1 + P_2 \equiv_\alpha Q_1 + Q_2}$$

$$\textsc{AlpRes1}\ \frac{P \equiv_\alpha Q \qquad x \neq y \qquad y \notin n(Q) \qquad x \in fn(Q)}{(\nu x)P \equiv_\alpha (\nu y)Q\{y/x\}}$$

$$\textsc{AlpInp1}\ \frac{P \equiv_\alpha Q \qquad x \neq y \qquad y \notin n(Q) \qquad x \in fn(Q)}{z(x).P \equiv_\alpha z(y).Q\{y/x\}}$$

$$\textsc{AlpRes2}\ \frac{P \equiv_\alpha Q \qquad x \neq y \qquad x \notin n(P) \qquad y \in fn(P)}{(\nu x)P\{x/y\} \equiv_\alpha (\nu y)Q}$$

$$\textsc{AlpInp2}\ \frac{P \equiv_\alpha Q\{x/y\} \qquad x \neq y \qquad x \notin n(P) \qquad y \in fn(P)}{z(x).P\{x/y\} \equiv_\alpha z(y).Q}$$

Table 2.6: α equivalence laws

Lemma 2.3.1. Let P be a process and y, w, z names such that $w = z$ or $w \notin fn(P)$ then $P\{w/z\}\{y/w\} \equiv_\alpha P$.

Definition 2.3.5. *structural congruence* $\equiv$ is the smallest relation on processes that satisfies the axioms in table 2.7

Proposition 2.3.2. $\equiv$ as defined in table 2.7 is a congruence and an equivalence relation.

Proof. $\equiv$ is a congruence thanks to rules $Cong1$ and $Cong2$. Reflexivity holds for rule *Alp*. Symmetry holds because all the rules are symmetric or have a symmetric counterpart. Transitivity holds because of rule $Trans$. □

We can make some clarification on the axioms of the structural congruence:

$unfolding$ this just helps replace an identifier by its definition, with the appropriate parameter instantiation. The alternative is to use the rule Cns in table 2.4.

α *conversion* is the α conversion, i.e., the choice of bound names, it identifies agents like $x(y).\overline{z}y$ and $x(w).\overline{z}w$. In the semantic of π calculus we can use the structural congruence with the rule *Alp* or we can embed the α conversion in the SOS rules. In the early case, the rule for input and the rules $ResAlp1, OpnAlp, Cns$ take care of α conversion, whether in the late case the rule for communication and the rules is $ResAlp1, OpnAlp, Cns$ are in charge for α conversion.

abelian monoidal properties of some operators We can deal with associativity and commutativity properties of sum and parallel composition by using SOS rules or by axiom of the structural congruence. For example the commutativity of the sum can be expressed by the following two rules:

$$\textbf{SumL}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \textbf{SumR}\ \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

SumAsc1 $M_1 + (M_2 + M_3) \equiv (M_1 + M_2) + M_3$ **ParAsc1** $P_1|(P_2|P_3) \equiv (P_1|P_2)|P_3$
SumAsc2 $(M_1 + M_2) + M_3 \equiv M_1 + (M_2 + M_3)$ **ParAsc2** $(P_1|P_2)|P_3 \equiv P_1|(P_2|P_3)$

ParCom $P_1|P_2 \equiv P_2|P_1$ **ResCom** $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$ **SumCom** $M_1 + M_2 \equiv M_2 + M_1$

ScpExtPar1 $\dfrac{z \notin fn(P_1)}{(\nu z)(P_1|P_2) \equiv P_1|(\nu z)P_2}$ **ScpExtPar2** $\dfrac{z \notin fn(P_1)}{P_1|(\nu z)P_2 \equiv (\nu z)(P_1|P_2)}$

ScpExtSum1 $\dfrac{z \notin fn(P_1)}{(\nu z)(P_1 + P_2) \equiv P_1 + (\nu z)P_2}$ **ScpExtSum2** $\dfrac{z \notin fn(P_1)}{P_1 + (\nu z)P_2 \equiv (\nu z)(P_1 + P_2)}$

Ide $\dfrac{A(\tilde{x}) \stackrel{def}{=} P}{A(\tilde{w}) \equiv P\{\tilde{w}/\tilde{x}\}}$ **Trans** $\dfrac{P \equiv Q \qquad Q \equiv R}{P \equiv R}$ **Alp** $\dfrac{P \equiv_\alpha Q}{P \equiv Q}$

Cong1 $\dfrac{P \equiv Q}{C[P] \equiv C[Q]}$ **Cong2** $\dfrac{P_1 \equiv Q_1 \qquad P_2 \equiv Q_2 \qquad C[\_,\_] \in \{\_ + \_, \_|\_\}}{C[P_1, P_2] \equiv C[Q_1, Q_2]}$

Table 2.7: Structural congruence rules

or by the following rule and axiom:

$$\textbf{Sum}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \textbf{SumCom} \quad P + Q \equiv Q + P$$

and the rule *Cong*

scope extension We can use the scope extension laws in table 2.7 or the rules *Opn* and *Cls* in table 2.4 to deal with the scope extension.

Lemma 2.3.3.
$$a \in fn(Q) \Rightarrow fn(Q\{b/a\}) = (fn(Q) - \{a\}) \cup \{b\}$$

Lemma 2.3.4. $P \equiv_\alpha Q \Rightarrow fn(P) = fn(Q)$

Proof. The proof goes by induction on rules

AlpZero the lemma holds because P and Q are the same process.

AlpTau :

	rule premise
$P \equiv_\alpha Q$	inductive hypothesis
$\Rightarrow fn(P) = fn(Q)$	definition of fn
$\Rightarrow fn(\tau.P) = fn(\tau.Q)$	

AlpOut :

	rule premise
$P \equiv_\alpha Q$	inductive hypothesis
$\Rightarrow fn(P) = fn(Q)$	
$\Rightarrow fn(P) \cup \{x, y\} = fn(Q) \cup \{x, y\}$	definition of fn
$\Rightarrow fn(\overline{x}y.P) = fn(\overline{x}y.Q)$	

*AlpRes*1 :

$$\begin{array}{ll}
 & \text{rule premise} \\
P \equiv_\alpha Q & \text{inductive hypothesis} \\
\Rightarrow fn(P) = fn(Q) & \\
\Rightarrow fn(P) - \{y\} = fn(Q) - \{y\} & \text{definition of } fn \\
\Rightarrow fn(P) - \{y\} = fn((\nu y)Q) & \\
\Rightarrow ((fn(P) - \{y\}) \cup \{x\}) - \{x\} = fn((\nu y)Q) & \\
\Rightarrow fn(P\{x/y\}) - \{x\} = fn((\nu y)Q) & \\
\Rightarrow fn((\nu x)(P\{x/y\})) = fn((\nu y)Q) &
\end{array}$$

$AlpRes2$: similar.

$AlpInp1$:

$$fn(a(x).(P\{x/y\})) = (fn(P\{x/y\}) - \{x\}) \cup \{a\} = (((fn(P) - \{y\}) \cup \{x\}) - \{x\}) \cup \{a\} = (fn(P) - \{y\}) \cup \{a\} = (fn(Q) - \{y\}) \cup \{a\} = fn(a(x).Q)$$

$AlpInp2$: similar.

$AlpSum$:

$$\begin{array}{ll}
 & \text{rule premises} \\
P_1 \equiv_\alpha Q_1 \text{ and } P_2 \equiv_\alpha Q_2 & \text{inductive hypothesis} \\
\Rightarrow fn(P_1) = fn(Q_1) \text{ and } fn(P_2) = fn(Q_2) & \\
\Rightarrow fn(P_1) \cup fn(P_2) = fn(Q_1) \cap fn(Q_2) & \text{definition of } fn \\
\Rightarrow fn(P_1 + P_2) = fn(Q_1 + Q_2) &
\end{array}$$

$AlpPar$:

$$\begin{array}{ll}
 & \text{rule premises} \\
P_1 \equiv_\alpha Q_1 \text{ and } P_2 \equiv_\alpha Q_2 & \text{inductive hypothesis} \\
\Rightarrow fn(P_1) = fn(Q_1) \text{ and } fn(P_2) = fn(Q_2) & \\
\Rightarrow fn(P_1) \cup fn(P_2) = fn(Q_1) \cap fn(Q_2) & \text{definition of } fn \\
\Rightarrow fn(P_1 | P_2) = fn(Q_1 | Q_2) &
\end{array}$$

$AlpRes$:

$$\begin{array}{ll}
 & \text{rule premise} \\
P \equiv_\alpha Q & \text{inductive hypothesis} \\
\Rightarrow fn(P) = fn(Q) & \\
\Rightarrow fn(P) - \{x\} = fn(Q) - \{x\} & \text{definition of } fn \\
\Rightarrow fn((\nu x)P) = fn((\nu x)Q) &
\end{array}$$

$AlpInp$:

$$\begin{array}{ll}
 & \text{rule premise} \\
P \equiv_\alpha Q\{x/y\} & \text{inductive hypothesis} \\
\Rightarrow fn(P) = fn(Q) & \\
\Rightarrow (fn(P) - \{y\}) \cup \{x\} = (fn(Q) - \{y\}) \cup \{x\} & \text{definition of } fn \\
\Rightarrow fn(x(y).P) = fn(x(y).Q) &
\end{array}$$

$AlpIde$ the lemma holds because P and Q are the same process.

□

Lemma 2.3.5. $P \equiv_\alpha P\{a/b\}\{b/a\}$

Lemma 2.3.6. α equivalence is invariant with respect to substitution. In other words

$$\begin{array}{l} P \equiv_\alpha Q \\ b \notin fn(P) \\ b \notin fn(Q) \end{array} \quad \Rightarrow \quad P\{b/a\} \equiv_\alpha Q\{b/a\}$$

Lemma 2.3.7.

$$P \equiv_\alpha P\{x/y\}\{y/x\}$$

In the proof of equivalence of the semantics in the next section we need the following lemmas

Lemma 2.3.8. The α equivalence is reflexive.

Proof. : We prove $P \equiv_\alpha P$ by structural induction on P:

0 :

$$\textsc{AlpZero}\ \frac{}{0 \equiv_\alpha 0}$$

$\tau.P_1$: for induction $P_1 \equiv_\alpha P_1$ so

$$\textsc{AlpTau}\ \frac{P_1 \equiv_\alpha P_1}{\tau.P_1 \equiv_\alpha \tau.P_1}$$

$x(y).P_1$: for induction $P_1 \equiv_\alpha P_1$ so

$$\textsc{AlpInp}\ \frac{P_1 \equiv_\alpha P_1}{x(y).P_1 \equiv_\alpha x(y).P_1}$$

$\overline{x}y.P_1$: for induction $P_1 \equiv_\alpha P_1$ so

$$\textsc{AlpOut}\ \frac{P_1 \equiv_\alpha P_1}{\overline{x}y.P_1 \equiv_\alpha \overline{x}y.P_1}$$

$P_1 + P_2$: for induction $P_1 \equiv_\alpha P_1$ and $P_2 \equiv_\alpha P_2$ so

$$\textsc{AlpSum}\ \frac{P_1 \equiv_\alpha P_1 \qquad P_2 \equiv_\alpha P_2}{P_1 + P_2 \equiv_\alpha P_1 + P_2}$$

$P_1|P_2$: for induction $P_1 \equiv_\alpha P_1$ and $P_2 \equiv_\alpha P_2$ so

$$\textsc{AlpPar}\ \frac{P_1 \equiv_\alpha P_1 \qquad P_2 \equiv_\alpha P_2}{P_1|P_2 \equiv_\alpha P_1|P_2}$$

$(\nu x)P_1$: for induction $P_1 \equiv_\alpha P_1$ so

$$\textsc{AlpRes}\ \frac{P_1 \equiv_\alpha P_1}{(\nu x)P_1 \equiv_\alpha (\nu x)P_1}$$

$A(\tilde{x})$:

$$\textsc{AlpIde}\ \frac{}{A(\tilde{x}) \equiv_\alpha A(\tilde{x})}$$

□

Lemma 2.3.9. α equivalence is symmetric.

Proof. Every rule is symmetric or it has a symmetric counterpart. □

Lemma 2.3.10. α equivalence is transitive.

Theorem 2.3.11. α equivalence is an equivalence relation.

Proof. Follows from lemmas 2.3.8, 2.3.9 and 2.3.10. □

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{ParL}\ \frac{P \xrightarrow{\alpha} P^{'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P^{'}|Q} \qquad \textbf{ParR}\ \frac{Q \xrightarrow{\alpha} Q^{'} \qquad bn(\alpha) \cap fn(P) = \emptyset}{P|Q \xrightarrow{\alpha} P|Q^{'}}$$

$$\textbf{SumL}\ \frac{P \xrightarrow{\alpha} P^{'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P + Q \xrightarrow{\alpha} P^{'}} \qquad \textbf{SumR}\ \frac{Q \xrightarrow{\alpha} Q^{'} \qquad bn(\alpha) \cap fn(P) = \emptyset}{P + Q \xrightarrow{\alpha} Q^{'}}$$

$$\textbf{Res}\ \frac{P \xrightarrow{\alpha} P^{'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P^{'}} \qquad \textbf{Alp}\ \frac{P \equiv_{\alpha} Q \qquad P \xrightarrow{\alpha} P^{'}}{Q \xrightarrow{\alpha} P^{'}}$$

$$\textbf{EComL}\ \frac{P \xrightarrow{xy} P^{'} \qquad Q \xrightarrow{\overline{x}y} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}|Q^{'}} \qquad \textbf{EComR}\ \frac{P \xrightarrow{\overline{x}y} P^{'} \qquad Q \xrightarrow{xy} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}|Q^{'}}$$

$$\textbf{ClsL}\ \frac{P \xrightarrow{\overline{x}(z)} P^{'} \qquad Q \xrightarrow{xz} Q^{'} \qquad z \notin fn(Q)}{P|Q \xrightarrow{\tau} (\nu z)(P^{'}|Q^{'})} \qquad \textbf{ClsR}\ \frac{P \xrightarrow{xz} P^{'} \qquad Q \xrightarrow{\overline{x}(z)} Q^{'} \qquad z \notin fn(P)}{P|Q \xrightarrow{\tau} (\nu z)(P^{'}|Q^{'})}$$

$$\textbf{Ide}\ \frac{A(\tilde{x}) \stackrel{def}{=} P \qquad P\{\tilde{w}/\tilde{x}\} \xrightarrow{\alpha} P^{'}}{A(\tilde{x}) \xrightarrow{\alpha} P^{'}} \qquad \textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P^{'} \qquad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P^{'}}$$

Table 2.8: Early transition relation with α conversion but without structural congruence

2.4 Operational semantic with structural congruence

2.4.1 Early semantic with α conversion only

In this subsection we introduce the early operational semantic for π calculus with the use of a minimal structural congruence, specifically we exploit only the easy of α conversion.

Definition 2.4.1. The *early transition relation with α conversion* $\rightarrow \subseteq \mathbb{P} \times \mathbb{A} \times \mathbb{P}$ is the smallest relation induced by the rules in table 2.8.

The following example shows why the condition $bn(\alpha) \cap fn(Q) = \emptyset$ in the rule Sum is desirable:

Example without the side condition we are able to prove:

$$\textbf{ClsL}\ \frac{\textbf{Sum}\ \dfrac{\textbf{Opn}\ \dfrac{(\nu y)\overline{x}y.0 \xrightarrow{\overline{x}y} (\nu y)0}{(\nu y)\overline{x}y.0 \xrightarrow{\overline{x}(y)} (\nu y)0}}{((\nu y)\overline{x}y.0) + \overline{y}x.0 \xrightarrow{\overline{x}(y)} (\nu y)0} \qquad \textbf{EInp}\ \dfrac{}{x(z).0 \xrightarrow{xy} 0}}{(((\nu y)\overline{x}y.0) + \overline{y}x.0)|x(z).0 \xrightarrow{\tau} (\nu y)0}$$

but $(((\nu y)\overline{x}y.0) + \overline{y}x.0)|x(z).0 \not\equiv (\nu y)(\overline{x}y.0 + \overline{y}x.0|x(z).0)$

2.4.2 Early semantic with structural congruence

Definition 2.4.2. The *early transition relation with structural congruence* $\rightarrow \subseteq \mathbb{P} \times \mathbb{A} \times \mathbb{P}$ is the smallest relation induced by the rules in table 2.9.

Example We prove now that

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{Par}\ \frac{P \xrightarrow{\alpha} P' \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \textbf{Sum}\ \frac{P \xrightarrow{\alpha} P' \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P + Q \xrightarrow{\alpha} P'}$$

$$\textbf{ECom}\ \frac{P \xrightarrow{xy} P' \qquad Q \xrightarrow{\overline{x}y} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad \textbf{Cong}\ \frac{P \equiv P' \qquad P' \xrightarrow{\alpha} Q}{P \xrightarrow{\alpha} Q}$$

$$\textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P' \qquad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P'} \qquad \textbf{Res}\ \frac{P \xrightarrow{\alpha} P' \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P'}$$

Table 2.9: Early semantic with structural congruence

$$a(x).P \mid (\nu b)\overline{a}b.Q \ \xrightarrow{\tau}\ (\nu b)(P\{b/x\} \mid Q)$$

where $b \notin fn(P)$. This follows from

$$a(x).P \mid (\nu b)\overline{a}b.Q \ \equiv\ (\nu b)(a(x).P \mid \overline{a}b.Q)$$

and

$$(\nu b)(a(x).P \mid \overline{a}b.Q) \xrightarrow{\tau} (\nu b)(P\{b/x\} \mid Q)$$

with the rule *Cong*. We can prove the last transition in the following way:

$$\textsc{Res}\ \dfrac{\textsc{Com}\ \dfrac{\textsc{EInp}\ \dfrac{}{a(x).P \ \xrightarrow{ab}\ P\{b/x\}} \qquad \textsc{Out}\ \dfrac{}{\overline{a}b.Q \ \xrightarrow{\overline{a}b}\ Q}}{a(x).P \mid \overline{a}b.Q \ \xrightarrow{\tau}\ P\{b/x\} \mid Q}}{(\nu b)(a(x).P \mid \overline{a}b.Q) \ \xrightarrow{\tau}\ (\nu b)(P\{b/x\} \mid Q)}$$

Example We want to prove now that:

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

where the name c is not in the free names of Q. We can exploit the structural congruence and get that

$$((\nu b)a(x).P)|\overline{a}b.Q \ \equiv\ (\nu c)(a(x).(P\{c/b\})|\overline{a}b.Q)$$

then we have

$$\textsc{Res}\ \dfrac{\textsc{Com}\ \dfrac{\textsc{EInp}\ \dfrac{}{a(x).P\{c/b\} \ \xrightarrow{ab}\ P\{c/b\}\{b/x\}} \qquad \textsc{Out}\ \dfrac{}{\overline{a}b.Q \ \xrightarrow{\overline{a}b}\ Q}}{(a(x).(P\{c/b\})|\overline{a}b.Q) \ \xrightarrow{\tau}\ (P\{c/b\}\{b/x\}|Q)}}{(\nu c)(a(x).(P\{c/b\})|\overline{a}b.Q) \ \xrightarrow{\tau}\ (\nu c)(P\{c/b\}\{b/x\}|Q)}$$

Now we just apply the rule *Cong* to prove the thesis.

2.4.3 Late semantic with structural congruence

Definition 2.4.3. The *late transition relation with structural congruence* $\rightarrow \subseteq \mathbb{P} \times \mathbb{A} \times \mathbb{P}$ is the smallest relation induced by the rules in table 2.10.

Example We prove now that

$$a(x).P \mid (\nu b)\overline{a}b.Q \ \xrightarrow{\tau} P\{b/x\} \mid Q$$

$$\textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textbf{LInp}\ \frac{}{x(y).P \xrightarrow{x(y)} P} \qquad \textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P}$$

$$\textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P' \qquad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P'} \qquad \textbf{Res}\ \frac{P \xrightarrow{\alpha} P' \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P'}$$

$$\textbf{LCom}\ \frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}z} Q'}{P|Q \xrightarrow{\tau} P'\{z/y\}|Q'} \qquad \textbf{Par}\ \frac{P \xrightarrow{\alpha} P' \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P'|Q}$$

$$\textbf{Cong}\ \frac{P \equiv P' \qquad P \xrightarrow{\alpha} Q}{P' \xrightarrow{\alpha} Q'} \qquad \textbf{Sum}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

Table 2.10: Late semantic with structural congruence

where $b \notin fn(P)$. This follows from

$$a(x).P \mid (\nu b)\overline{a}b.Q \;\equiv\; (\nu b)(a(x).P \mid \overline{a}b.Q)$$

and

$$(\nu b)(a(x).P \mid \overline{a}b.Q) \xrightarrow{\tau} (\nu b)(P\{b/x\} \mid Q)$$

with the rule *Cong*. We can prove the last transition in the following way:

$$\textsc{Res}\ \frac{\textsc{LCom}\ \dfrac{\textsc{LInp}\ \dfrac{b \notin fn(P)}{a(x).P \xrightarrow{ab} P\{b/x\}} \qquad \textsc{Out}\ \dfrac{}{\overline{a}b.Q \xrightarrow{\overline{a}b} Q}}{a(x).P \mid \overline{a}b.Q \xrightarrow{\tau} P\{b/x\} \mid Q} \qquad b \notin n(\tau)}{(\nu b)(a(x).P \mid \overline{a}b.Q) \xrightarrow{\tau} (\nu b)(P\{b/x\} \mid Q)}$$

Example We want to prove now that:

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

where the name c is not in the free names of Q and is not in the names of P. We can exploit the structural congruence and get that

$$((\nu b)a(x).P)|\overline{a}b.Q \;\equiv\; (\nu c)(a(x).(P\{c/b\})|\overline{a}b.Q)$$

then we have

$$\textsc{Res}\ \frac{\textsc{LCom}\ \dfrac{\textsc{LInp}\ \dfrac{b \notin fn(P\{c/b\})}{a(x).P\{c/b\} \xrightarrow{ab} P\{c/b\}\{b/x\}} \qquad \textsc{Out}\ \dfrac{}{\overline{a}b.Q \xrightarrow{\overline{a}b} Q}}{(a(x).(P\{c/b\})|\overline{a}b.Q) \xrightarrow{\tau} (P\{c/b\}\{b/x\}|Q)} \qquad c \notin n(\tau)}{(\nu c)(a(x).(P\{c/b\})|\overline{a}b.Q) \xrightarrow{\tau} (\nu c)(P\{c/b\}\{b/x\}|Q)}$$

Now we just apply the rule *Cong* to prove the thesis.

2.5 Equivalence of the semantics

2.5.1 Equivalence of the early semantics

In this subsection we write $\rightarrow_1$ for the early semantic without structural congruence, $\rightarrow_2$ for the early semantic with just α conversion and $\rightarrow_3$ for the early semantic with the full structural congruence. We call R_1 the set of rules for $\rightarrow_1$, R_2 the set of rules for $\rightarrow_2$ and R_3 the set of rules for $\rightarrow_3$.

Lemma 2.5.1. Structurally equivalent process have the same free names:

$$P \equiv Q \;\Rightarrow\; fn(Q) = fn(P)$$

Proof. The proof is easy and is an induction on the rules of structural congruence. $\square$

We would like to prove that $P \xrightarrow{\alpha}_2 P' \;\Rightarrow\; P \xrightarrow{\alpha}_1 P'$ but this is false because

$$\textsc{Alp}\;\dfrac{\overline{x}y.x(y).0 \equiv_\alpha \overline{x}y.x(w).0 \qquad \textsc{Out}\;\dfrac{}{\overline{x}y.x(w).0 \xrightarrow{\overline{x}y}_2 x(w).0}}{\overline{x}y.x(y).0 \xrightarrow{\overline{x}y}_2 x(w).0}$$

so we want to prove

$$\overline{x}y.x(y).0 \xrightarrow{\overline{x}y}_1 x(w).0$$

The head of the transition has an output prefixing at the top level so the only rule we could use is *Out*, but the application of *Out* yields

$$\overline{x}y.x(y).0 \xrightarrow{\overline{x}y}_1 x(y).0$$

which is not want we want. But we prove some weaker results.

Lemma 2.5.2. Let $\twoheadrightarrow_2$ be the semantic of table 2.8 but without rule *Alp*. If $P \xrightarrow{\alpha}_2 P'$ then there exist a process Q such that $P \equiv_\alpha Q \overset{\alpha}{\twoheadrightarrow}_2 P'$.

Proof. We prove by cases that in a derivation of $P \xrightarrow{\alpha}_2 P'$ we can move downward to the end of the derivation any occurrence of the rule *Alp*:

$ParL$:

$$\textbf{ParL}\;\dfrac{\textbf{Alp}\;\dfrac{P \equiv_\alpha R \qquad R \xrightarrow{\alpha} P'}{P \xrightarrow{\alpha} P'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P'}$$

became

$$\textbf{Alp}\;\dfrac{\textbf{AlpPar}\;\dfrac{P \equiv_\alpha R}{P|Q \equiv_\alpha R|Q} \qquad \textbf{ParL}\;\dfrac{R \xrightarrow{\alpha} P' \qquad bn(\alpha) \cap fn(Q) = \emptyset}{R|Q \xrightarrow{\alpha} P'}}{P|Q \xrightarrow{\alpha} P'}$$

$ParR, SumL, SumR$ similar.

Alp since α equivalence is transitive, we can merge any pair of consecutive instance of the rule *Alp*

Res :

$$\textbf{Res}\;\dfrac{\textbf{Alp}\;\dfrac{P \equiv_\alpha R \qquad R \xrightarrow{\alpha} P'}{P \xrightarrow{\alpha} P'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P'}$$

became

$$\textbf{Alp}\;\dfrac{\textbf{AlpRes}\;\dfrac{P \equiv_\alpha R}{(\nu z)P \equiv_\alpha (\nu z)R} \qquad \textbf{Res}\;\dfrac{R \xrightarrow{\alpha} P' \qquad z \notin n(\alpha)}{(\nu z)R \xrightarrow{\alpha} (\nu z)P'}}{(\nu z)P \xrightarrow{\alpha} (\nu z)P'}$$

Opn similar.

$EComL$:

$$\textbf{EComL}\ \frac{\textbf{Alp}\ \dfrac{P \equiv_\alpha R \qquad R \xrightarrow{xy} P'}{P \xrightarrow{xy} P'} \qquad \textbf{Alp}\ \dfrac{Q \equiv_\alpha S \qquad S \xrightarrow{\overline{x}y} S'}{Q \xrightarrow{\overline{x}y} S'}}{P|Q \xrightarrow{\tau} P'|S'}$$

became

$$\textbf{Alp}\ \frac{\textbf{AlpPar}\ \dfrac{P \equiv_\alpha R \qquad Q \equiv_\alpha S}{P|Q \equiv_\alpha R|S} \qquad \textbf{EComL}\ \dfrac{R \xrightarrow{xy} P' \qquad S \xrightarrow{\overline{x}y} S'}{R|S \xrightarrow{\tau} S'}}{P|Q \xrightarrow{\tau} P'|S'}$$

$EComR$ similar.

$ClsR$:

$$\textbf{ClsR}\ \frac{\textbf{Alp}\ \dfrac{P \equiv_\alpha R \qquad R \xrightarrow{xy} P'}{P \xrightarrow{xy} P'} \qquad \textbf{Alp}\ \dfrac{Q \equiv_\alpha S \qquad S \xrightarrow{\overline{x}(y)} S'}{Q \xrightarrow{\overline{x}(y)} S'} \qquad z \notin fn(P)}{P|Q \xrightarrow{\tau} (\nu z)(P'|S')}$$

became

$$\textbf{Alp}\ \frac{\textbf{AlpPar}\ \dfrac{P \equiv_\alpha R \qquad Q \equiv_\alpha S}{P|Q \equiv_\alpha R|S} \qquad \textbf{ClsR}\ \dfrac{R \xrightarrow{xy} P' \qquad S \xrightarrow{\overline{x}y} S' \qquad z \notin fn(R)}{R|S \xrightarrow{\tau} (\nu y)(P'|S')}}{P|Q \xrightarrow{\tau} (\nu y)(P'|S')}$$

$ClsL$ similar.

Ide :

$$\textbf{Ide}\ \frac{A(\tilde{x}) \overset{def}{=} P \qquad \textbf{Alp}\ \dfrac{P\{\tilde{w}/\tilde{x}\} \equiv_\alpha R \qquad R \xrightarrow{\alpha} P'}{P\{\tilde{w}/\tilde{x}\} \xrightarrow{\alpha} P'}}{A(\tilde{w}) \xrightarrow{\alpha} P'}$$

we can add a new definition $A(\tilde{w}) \overset{def}{=} R$ and this derivation became:

$$\textbf{Ide}\ \frac{A(\tilde{w}) \overset{def}{=} R \qquad R \xrightarrow{\alpha} P'}{A(\tilde{w}) \xrightarrow{\alpha} P'}$$

□

Theorem 2.5.3. If $P \xrightarrow{\alpha}_2 P'$ then there exists a process Q such that $P \equiv_\alpha Q \xrightarrow{\alpha}_1 P'$.

Proof. This result follows from lemma 2.5.2 observing that $\twoheadrightarrow_2 \subseteq \rightarrow_1$. □

Theorem 2.5.4. If $P \xrightarrow{\alpha}_2 P'$ then there exists a process P'' such that $P \xrightarrow{\alpha}_1 P''$ and $P'' \equiv_\alpha P'$

Proof. For lemma 2.5.2 there exists a process Q such that $P \equiv_\alpha Q \overset{\alpha}{\twoheadrightarrow}_2 P'$. The proof proceed by cases on the last rule used in the derivation of $P \equiv_\alpha Q$:

$AlpIde$ in this case there is an identifier such that $P = A(\tilde{x}) = Q$ so the conclusion holds.

$AlpInp$ in this case $P = x(y).P_1$, $Q = x(y).Q_1$, $P_1 \equiv_\alpha Q_1$ and $\alpha = xz$. For rule $EInp$: $x(y).Q_1 \xrightarrow{xz} Q_1$.

*AlpInp*1 :

$$\textbf{Alp}\ \dfrac{\textbf{AlpInp1}\ \dfrac{P_1 \equiv_\alpha Q_1}{a(x).P_1 \equiv_\alpha a(y).(Q_1\{y/x\})} \qquad \textbf{EInp}\ \dfrac{}{a(y).Q_1\{y/x\} \xrightarrow{az}_2 Q_1\{y/x\}\{z/y\}}}{a(x).P_1 \xrightarrow{az}_2 Q_1\{y/x\}\{z/y\}}$$

For rule *EInp*: $a(x).P_1 \xrightarrow{az}_1 P_1$. $P_1 \equiv_\alpha Q_1$ imply $P_1 \equiv_\alpha Q_1\{y/x\}\{z/y\}$

*AlpInp*2 :

$$\textbf{Alp}\ \dfrac{\textbf{AlpInp2}\ \dfrac{P_1 \equiv_\alpha Q_1}{a(x).(P_1\{x/y\}) \equiv_\alpha a(y).Q_1} \qquad \textbf{EInp}\ \dfrac{}{a(y).Q_1 \xrightarrow{az}_2 Q_1\{z/y\}}}{a(x).(P_1\{x/y\}) \xrightarrow{az}_2 Q_1\{z/y\}}$$

For rule *EInp*: $a(x).(P_1\{x/y\}) \xrightarrow{az}_1 P_1\{x/y\}\{z/x\}$. $P_1 \equiv_\alpha Q_1$ imply $P_1\{x/y\}\{z/x\} \equiv_\alpha Q_1\{z/y\}$

AlpOut in this case $P = \overline{a}x.P_1$, $Q = \overline{a}x.Q_1$, $P_1 \equiv_\alpha Q_1$ and $\alpha = \overline{a}x$. For rule *Out*: $\overline{a}x.P_1 \xrightarrow{\overline{a}x} P_1$ and $\overline{a}x.Q_1 \xrightarrow{\overline{a}x} Q_1$.

AlpTau similar.

AlpPar in this case $P = P_1|P_2$, $Q = Q_1|Q_2$, $P_1 \equiv_\alpha Q_1$ and $P_2 \equiv_\alpha Q_2$. The last rule used in the derivation of $Q_1|Q_2 \overset{\alpha}{\twoheadrightarrow}_2 P^{'}|Q_2$ can be:

- *ParL* in this case $P_1 \equiv Q_1 \overset{\alpha}{\twoheadrightarrow}_2 P^{'}$ and for inductive hypothesis $P_1 \xrightarrow{\alpha}_1 P^{'}$. For rule *ParL*: $P_1|P_2 \xrightarrow{\alpha}_1 P^{'}|P_2 \equiv_\alpha P^{'}|Q_2$
- *ParR* similar.
- *EComL* in this case $P_1 \equiv Q_1 \overset{xy}{\twoheadrightarrow}_2 Q_1^{'}$ and $P_2 \equiv Q_2 \overset{\overline{x}y}{\twoheadrightarrow}_2 Q_2^{'}$, for inductive hypothesis $P_1 \xrightarrow{xy}_1 Q_1^{'}$ and $P_2 \xrightarrow{\overline{x}y}_1 Q_2^{'}$. For rule *EComL*: $P_1|P_2 \xrightarrow{\tau}_1 Q_1^{'}|Q_2^{'}$
- *EComR* similar.

AlpSum similar.

AlpRes in this case $P = (\nu x)P_1$, $Q = (\nu x)Q_1$ and $P_1 \equiv Q_1$. The last rule used in the derivation of $Q \overset{\alpha}{\twoheadrightarrow}_2 P^{'}$ can be:

- *Res* in this case $P_1 \equiv Q_1 \overset{\alpha}{\twoheadrightarrow}_2 Q_1^{'}$ and $x \notin n(\alpha)$. For inductive hypothesis $P_1 \xrightarrow{\alpha}_1 Q_1^{'}$. For rule *Res*: $(\nu x)P_1 \xrightarrow{\alpha}_1 (\nu x)Q_1^{'}$
- *Opn* similar.

*AlpRes*1 The last rule used in the derivation of $Q \overset{\alpha}{\twoheadrightarrow}_2 P^{'}$ can be:

Res :

$$\textbf{Alp}\ \dfrac{\textbf{AlpRes1}\ \dfrac{P_1 \equiv_\alpha Q_1}{(\nu x)P_1 \equiv_\alpha (\nu y)(Q_1\{y/x\})} \qquad \textbf{Res}\ \dfrac{Q_1\{y/x\} \xrightarrow{\alpha}_2 Q^{'} \qquad y \notin n(\alpha)}{(\nu y)(Q_1\{y/x\}) \xrightarrow{\alpha}_2 (\nu y)Q^{'}}}{(\nu x)P_1 \xrightarrow{\alpha}_2 (\nu y)Q^{'}}$$

$P_1 \equiv_\alpha Q_1$ imply $P_1\{y/x\} \equiv_\alpha Q_1\{y/x\}$, for inductive hypothesis $P_1\{y/x\} \xrightarrow{\alpha}_1 Q^{'}$, now we can derive

$$\textbf{ResAlp1}\ \dfrac{\textbf{Res}\ \dfrac{P_1\{y/x\} \xrightarrow{\alpha}_1 Q^{'} \qquad y \notin n(\alpha)}{(\nu y)(P_1\{y/x\}) \xrightarrow{\alpha}_1 (\nu y)Q^{'}}}{(\nu x)P_1 \xrightarrow{\alpha}_1 (\nu y)Q_1^{'}}$$

Opn similar.

*AlpRes*2 The last rule used in the derivation of $Q \xrightarrow{\alpha}_2 P'$ can be:

Res :

$$\textbf{Alp}\ \frac{\textbf{AlpRes2}\ \dfrac{P_1 \equiv_\alpha Q_1}{(\nu x)(P_1\{x/y\}) \equiv_\alpha (\nu y)Q_1} \qquad \textbf{Res}\ \dfrac{Q_1 \xrightarrow{\alpha}_2 Q' \qquad y \notin n(\alpha)}{(\nu y)Q_1 \xrightarrow{\alpha}_2 (\nu y)Q'}}{(\nu x)(P_1\{x/y\}) \xrightarrow{\alpha}_2 (\nu y)Q'}$$

$P_1 \equiv_\alpha Q_1 \xrightarrow{\alpha}_2 Q'$ imply for inductive hypothesis that $P_1 \xrightarrow{\alpha}_1 Q'$, now we can derive

$$\textbf{ResAlp2}\ \frac{P_1 \xrightarrow{\alpha}_1 Q' \qquad y \notin n(\alpha)}{(\nu x)P_1\{x/y\} \xrightarrow{\alpha}_1 (\nu y)Q'}$$

Opn similar.

AlpZero this case does not exist.

□

Theorem 2.5.5. $P \xrightarrow{\alpha}_1 P' \Rightarrow P \xrightarrow{\alpha}_2 P'$

Proof. The proof can go by induction on the length of the derivation of a transaction, and then both the base case and the inductive case proceed by cases on the last rule used in the derivation. However it is not necessary to show all the details of the proof because the rules in R_2 are almost the same as the rules in R_1, the only difference is that in R_2 we have the rule *Alp* instead of *ResAlp*1, *ResAlp*2 and *OpnAlp*. The rule *Alp* can mimic the rule *ResAlp*1 in the following way:

$$\textbf{Alp}\ \frac{\textbf{AlpRes1}\ \dfrac{w \notin n(P)}{(\nu z)P \equiv_\alpha (\nu w)P\{w/z\}} \qquad (\nu w)P\{w/z\} \xrightarrow{xz} P'}{(\nu z)P \xrightarrow{xz} P'}$$

The rule *Alp* can mimic the rule *ResAlp*2 in the following way:

$$\textbf{Alp}\ \frac{\textbf{AlpRes2}\ \dfrac{w \notin n(P)}{(\nu w)P\{w/z\} \equiv_\alpha (\nu z)P} \qquad \textbf{Alp}\ \dfrac{P \xrightarrow{\alpha} P' \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha} (\nu z)P'}}{(\nu w)P\{w/z\} \xrightarrow{\alpha} (\nu z)P'}$$

The rule *Alp* can mimic the rule *OpnAlp* in the following way:

$$\textbf{Alp}\ \frac{\textbf{AlpRes1}\ \dfrac{w \notin n(P)}{(\nu z)P \equiv_\alpha (\nu w)P\{w/z\}} \qquad (\nu w)P\{w/z\} \xrightarrow{\overline{x}(w)} P'}{(\nu z)P \xrightarrow{\overline{x}(w)} P'}$$

□

Lemma 2.5.6. If $P \xrightarrow{\overline{x}(y)}_2 P'$ then there is a process R such that $P \equiv R \xrightarrow{\overline{x}(y)}_2 P'$ and the last rule in this derivation is the instance of rule *Opn* used to open the scope of y.

Proof. The derivation of $P \xrightarrow{\overline{x}(y)}_2 P'$ must contain an instance of *Opn*. The proof consists in showing that we can move this instance of *Opn* downward in the inference tree of $P \xrightarrow{\overline{x}(y)}_2 P'$. The proof goes by induction on the depth of the derivation of $P \xrightarrow{\overline{x}(y)}_2 P'$ and then by cases on the last rule applied:

Opn if the derivation ends with *Opn* then the conclusion holds.

SumL :

$$\textbf{SumL}\ \dfrac{\textbf{Opn}\ \dfrac{P_1 \xrightarrow{\overline{x}y}_2 P' \qquad x \neq y}{(\nu y)P_1 \xrightarrow{\overline{x}(y)}_2 P'} \qquad bn(\overline{x}(y)) \cap fn(R) = \emptyset}{P = (\nu y)P_1 + R \xrightarrow{\overline{x}(y)}_2 P'}$$

became:

$$\textbf{Opn}\ \dfrac{\textbf{SumL}\ \dfrac{P_1 \xrightarrow{\overline{x}y}_2 P'}{P_1 + R \xrightarrow{\overline{x}y}_2 P'} \qquad x \neq y}{(\nu y)(P_1 + R) \xrightarrow{\overline{x}(y)}_2 P'}$$

$bn(\overline{x}(y)) \cap fn(R) = \emptyset$ imply $y \notin fn(R)$ and so $(\nu y)(P_1 + R) \equiv (\nu y)P_1 + R$.

$SumR$ symmetric to the previous case.

$ParL$:

$$\textbf{ParL}\ \dfrac{\textbf{Opn}\ \dfrac{P_1 \xrightarrow{\overline{x}y}_2 P' \qquad x \neq y}{(\nu y)P_1 \xrightarrow{\overline{x}(y)}_2 P'} \qquad bn(\overline{x}(y)) \cap fn(R) = \emptyset}{P = (\nu y)P_1|R \xrightarrow{\overline{x}(y)}_2 P'|R}$$

became:

$$\textbf{Opn}\ \dfrac{\textbf{ParL}\ \dfrac{P_1 \xrightarrow{\overline{x}y}_2 P'}{P_1|R \xrightarrow{\overline{x}y}_2 P'|R} \qquad x \neq y}{(\nu y)(P_1|R) \xrightarrow{\overline{x}(y)}_2 P'|R}$$

$bn(\overline{x}(y)) \cap fn(R) = \emptyset$ imply $y \notin fn(R)$ and so $(\nu y)(P_1|R) \equiv (\nu y)P_1|R$.

$ParR$ symmetric to the previous case.

Res :

$$\textbf{Res}\ \dfrac{\textbf{Opn}\ \dfrac{P_1 \xrightarrow{\overline{x}y}_2 P' \qquad x \neq y}{(\nu y)P_1 \xrightarrow{\overline{x}(y)}_2 P'} \qquad w \notin n(\overline{x}(y))}{P = (\nu w)(\nu y)P_1 \xrightarrow{\overline{x}(y)}_2 (\nu w)P'}$$

became:

$$\textbf{Opn}\ \dfrac{\textbf{Res}\ \dfrac{P_1 \xrightarrow{\overline{x}y}_2 P' \qquad w \notin n(\overline{x}y)}{(\nu w)P_1 \xrightarrow{\overline{x}y}_2 (\nu w)P'} \qquad x \neq y}{(\nu y)(\nu w)P_1 \xrightarrow{\overline{x}(y)}_2 (\nu w)P'}$$

$(\nu y)(\nu w)P_1 \equiv (\nu w)(\nu y)P_1$.

$Alp(1)$:

$$\textbf{Alp}\ \dfrac{\dfrac{P_1 \equiv_\alpha R_1}{(\nu y)P_1 \equiv_\alpha (\nu y)R_1} \qquad \textbf{Opn}\ \dfrac{R_1 \xrightarrow{\overline{x}y}_2 R_1' \qquad x \neq y}{(\nu y)R_1 \xrightarrow{\overline{x}(y)}_2 R_1'}}{P = (\nu y)P_1 \xrightarrow{\overline{x}(y)}_2 R_1' = P'}$$

became:

$$\textbf{Opn}\ \dfrac{\textbf{Alp}\ \dfrac{P_1 \equiv_\alpha R_1 \qquad R_1 \xrightarrow{\overline{x}y}_2 R_1^{'}}{P_1 \xrightarrow{\overline{x}y}_2 R_1^{'}} \qquad x \neq y}{(\nu y)P_1 \xrightarrow{\overline{x}(y)}_2 R_1^{'}}$$

$Alp(2)$:

$$\textbf{Alp}\ \dfrac{\dfrac{P_1\{w/y\} \equiv_\alpha R_1 \qquad w \notin n(P_1)}{(\nu w)P_1 \equiv_\alpha (\nu y)R_1} \qquad \textbf{Opn}\ \dfrac{R_1 \xrightarrow{\overline{x}y}_2 R_1^{'} \qquad x \neq y}{(\nu y)R_1 \xrightarrow{\overline{x}(y)}_2 R_1^{'}}}{P = (\nu w)P_1 \xrightarrow{\overline{x}(y)}_2 R_1^{'} = P^{'}}$$

became:

$$\textbf{Opn}\ \dfrac{\textbf{Alp}\ \dfrac{P_1\{y/w\} \equiv_\alpha R_1 \qquad R_1 \xrightarrow{\overline{x}y}_2 R_1^{'}}{P_1\{y/w\} \xrightarrow{\overline{x}y}_2 R_1^{'}} \qquad x \neq y}{(\nu y)P_1\{y/w\} \xrightarrow{\overline{x}(y)}_2 R_1^{'}}$$

and $(\nu y)P_1\{y/w\} \equiv (\nu w)P_1$

□

Lemma 2.5.7. $P \xrightarrow{\alpha}_2 P^{'}$ imply that there exist processes $Q, Q^{'}$ such that $P \equiv Q \xrightarrow{\alpha}_3 Q^{'} \equiv P^{'}$

Proof. The proof is by induction on the length of the derivation of $P \xrightarrow{\alpha}_2 P^{'}$ and then both the base case and the inductive case proceed by cases on the last rule used:

$Out, EInp, Tau$ in this case the rule used can be one of the following $Out, EInp, Tau$ which are also in R_3 so a derivation of $P \xrightarrow{\alpha}_2 P^{'}$ is also a derivation of $P \xrightarrow{\alpha}_3 P^{'}$

Res, Opn the last rule used can be one in $R_2 \cap R_3 = \{Res, Opn\}$ and so for example we have

$$\textbf{Res}\ \dfrac{P \xrightarrow{\alpha}_2 P^{'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha}_2 (\nu z)P^{'}}$$

we apply the inductive hypothesis on $P \xrightarrow{\alpha}_2 P^{'}$ and get that there exists a process $P^{''}$ such that $P^{'} \equiv P^{''}$ and $P \xrightarrow{\alpha}_3 P^{''}$. The proof we want is:

$$\textbf{Res}\ \dfrac{P \xrightarrow{\alpha}_3 P^{''} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\alpha}_3 (\nu z)P^{''}}$$

and $(\nu z)P^{''} \equiv (\nu z)P^{'}$

$ParL, ParR, SumL, SumR, EComL, EComR$ In this cases we can proceed as in the previous case and if necessary add an application of $Cong$ thus exploiting the commutativity of sum or parallel composition. For example

$$\textbf{ParR}\ \dfrac{Q \xrightarrow{\alpha}_2 Q^{'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha}_2 P|Q^{'}}$$

now we apply the inductive hypothesis to $Q \xrightarrow{\alpha}_2 Q^{'}$ and get $Q \xrightarrow{\alpha}_3 Q^{''}$ for a $Q^{''}$ such that $Q^{'} \equiv Q^{''}$. The proof we want is

$$\textbf{Cong}\ \dfrac{P|Q \equiv Q|P \qquad \textbf{Par}\ \dfrac{Q \xrightarrow{\alpha}_3 Q^{''} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{Q|P \xrightarrow{\alpha}_3 Q^{''}|P}}{P|Q \xrightarrow{\alpha}_3 Q^{''}|P}$$

and $Q^{''}|P \equiv P|Q^{'}$

Ide if the last rule used is *Ide*:

$$\textbf{Ide}\ \frac{A(\tilde{x}) \stackrel{def}{=} P \qquad P\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha}_2 P'}{A(\tilde{y}) \xrightarrow{\alpha}_2 P'}$$

we apply the inductive hypothesis on the premise and get that there exists a process P'' such that $P\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha}_3 P''$ and $P'' \equiv P'$. Now the proof we want is

$$\textbf{Ide}\ \frac{A(\tilde{y}) \equiv P\{\tilde{y}/\tilde{x}\} \qquad P\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha}_3 P''}{A(\tilde{y}) \xrightarrow{\alpha}_3 P''}$$

Alp the rule *Alp* is a particular case of the rule *Cong*

ClsL if the last rule is *ClsL* then we have

$$\textbf{ClsL}\ \frac{P \xrightarrow{\overline{x}(z)}_2 P' \qquad Q \xrightarrow{xz}_2 Q' \qquad z \notin fn(Q)}{P|Q \xrightarrow{\tau}_2 (\nu z)(P'|Q')}$$

for lemma 2.5.6: $P \xrightarrow{\overline{x}(z)}_2 P'$ imply that there exist a process $(\nu z)R$ such that $P \equiv (\nu z)R$, $(\nu z)R \xrightarrow{\overline{x}(z)}_2 P'$ and the derivation of $(\nu z)R \xrightarrow{\tau}_2 R'$ ends with the instance of *Opn* that opens the scope of z. So

$$\textbf{Res}\ \frac{\textbf{EComL}\ \dfrac{R \xrightarrow{\overline{x}z}_2 P' \qquad Q \xrightarrow{xz}_2 Q'}{R|Q \xrightarrow{\tau}_2 P'|Q'}}{(\nu z)(R|Q) \xrightarrow{\tau}_2 (\nu z)(P'|Q')}$$

$P \equiv (\nu z)R$ and $z \notin fn(Q)$ imply $(\nu z)(R|Q) \equiv P|Q$. The conclusion follows after applying the inductive hypothesis on $(\nu z)(R|Q) \xrightarrow{\tau}_3 (\nu z)(P'|Q')$ and the transitivity of structural congruence.

ClsR symmetric.

□

Theorem 2.5.8. $P \xrightarrow{\alpha}_2 P'$ imply that there exist processes Q' such that $P \xrightarrow{\alpha}_3 Q' \equiv P'$.

Proof. For lemma 2.5.7 there exist processes Q, Q' such that $P \equiv Q \xrightarrow{\alpha}_3 Q' \equiv P'$. So for rule *Cong*: $P \xrightarrow{\alpha}_3 Q' \equiv P'$. □

Lemma 2.5.9. Let $\twoheadrightarrow_3$ be the semantic in table 2.9 but without rule *Cong*. $P \xrightarrow{\alpha}_3 P'$ imply that there exist a process Q such that $P \equiv Q \overset{\alpha}{\twoheadrightarrow}_3 P'$.

Proof. The proof needs to show that in any proof tree we can move downward any instance of a rule *Cong* until the proof tree has only on instance of the rule *Cong* and this is at the end. There are some cases to consider:

Sum :

$$\textbf{Sum}\ \frac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\alpha} P'}{P \xrightarrow{\alpha} P'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P + Q \xrightarrow{\alpha} P'}$$

became:

$$\textbf{Cong}\ \frac{\dfrac{P \equiv R}{P + Q \equiv R + Q} \qquad \textbf{Sum}\ \dfrac{R \xrightarrow{\alpha} P' \qquad bn(\alpha) \cap fn(Q) = \emptyset}{R + Q \xrightarrow{\alpha} P'}}{P + Q \xrightarrow{\alpha} P'}$$

Par :

$$\textbf{Par}\ \dfrac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\alpha} P^{'}}{P \xrightarrow{\alpha} P^{'}} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P^{'}|Q}$$

became:

$$\textbf{Cong}\ \dfrac{\dfrac{P \equiv R}{P|Q \equiv R|Q} \qquad \textbf{Par}\ \dfrac{R \xrightarrow{\alpha} P^{'} \qquad bn(\alpha) \cap fn(Q) = \emptyset}{R|Q \xrightarrow{\alpha} P^{'}|Q}}{P + Q \xrightarrow{\alpha} P^{'}|Q}$$

ECom :

$$\textbf{ECom}\ \dfrac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{xy} P^{'}}{P \xrightarrow{xy} P^{'}} \qquad Q \xrightarrow{\overline{x}y} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}|Q^{'}}$$

became:

$$\textbf{Cong}\ \dfrac{\dfrac{P \equiv R}{P|Q \equiv R|Q} \qquad \textbf{ECom}\ \dfrac{R \xrightarrow{xy} P^{'} \qquad Q \xrightarrow{\overline{x}y} Q^{'}}{R|Q \xrightarrow{\tau} P^{'}|Q^{'}}}{P|Q \xrightarrow{\alpha} P^{'}|Q^{'}}$$

Res :

$$\textbf{Res}\ \dfrac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\alpha} P^{'}}{P \xrightarrow{\alpha} P^{'}} \qquad x \notin n(\alpha)}{(\nu x)P \xrightarrow{\alpha} (\nu x)P^{'}}$$

became:

$$\textbf{Cong}\ \dfrac{\dfrac{P \equiv R}{(\nu x)P \equiv (\nu x)R} \qquad \textbf{Res}\ \dfrac{R \xrightarrow{\alpha} P^{'} \qquad x \notin n(\alpha)}{(\nu x)R \xrightarrow{\alpha} (\nu x)P^{'}}}{(\nu x)P \xrightarrow{\alpha} (\nu x)P^{'}}$$

Opn :

$$\textbf{Opn}\ \dfrac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\overline{y}x} P^{'}}{P \xrightarrow{\overline{y}x} P^{'}}}{(\nu x)P \xrightarrow{\overline{y}(x)} P^{'}}$$

became:

$$\textbf{Cong}\ \dfrac{\dfrac{P \equiv R}{(\nu x)P \equiv (\nu x)R} \qquad \textbf{Opn}\ \dfrac{R \xrightarrow{\overline{y}x} P^{'}}{(\nu x)R \xrightarrow{\overline{y}(x)} P^{'}}}{(\nu x)P \xrightarrow{\overline{y}(x)} P^{'}}$$

Cong :

$$\textbf{Cong}\ \frac{P \equiv R \qquad \textbf{Cong}\ \dfrac{R \equiv S \qquad S \xrightarrow{\alpha} P^{'}}{R \xrightarrow{\alpha} P^{'}}}{P \xrightarrow{\alpha} P^{'}}$$

became:

$$\textbf{Cong}\ \frac{\dfrac{P \equiv R \qquad R \equiv S}{P \equiv S} \qquad S \xrightarrow{\alpha} P^{'}}{P \xrightarrow{\alpha} P^{'}}$$

□

Theorem 2.5.10. $P \xrightarrow{\alpha}_3 P^{'}$ imply that there exist a proces Q such that $P \equiv Q \xrightarrow{\alpha}_2 P^{'}$.

Proof. This is a direct consequence of lemma 2.5.9 observing that $\twoheadrightarrow_3 \subseteq \rightarrow_2$. □

2.5.2 Equivalence of the late semantics

2.6 Bisimilarity, congruence and equivalence

We present here some behavioural equivalences and some of their properties. In the following we will use the phrase $bn(\alpha)$ is fresh in a definition to mean that the name in $bn(\alpha)$, if any, is different from any free name occurring in any of the agents in the definition. We write $\rightarrow_E$ for the early semantic and $\rightarrow_L$ for the late semantic. It's not a concern which late semantic we are talking about because we have proved them equivalent.

2.6.1 Late bisimilarity

Definition 2.6.1. A *strong late bisimulation*(according to [4]) is a binary symmetric relation $\mathbf{S}$ on processes such that for each process P and Q, $P\mathbf{S}Q$ implies:

- if $P \xrightarrow{a(x)}_L P^{'}$ and $x \notin fn(P) \cup fn(Q)$ then there exists a process $Q^{'}$ such that $Q \xrightarrow{a(x)}_L Q^{'}$ and for all u $P^{'}\{u/x\}\mathbf{S}Q^{'}\{u/x\}$
- if $P \xrightarrow{\alpha}_L P^{'}$, α is not an input and $bn(\alpha) \cap (fn(P) \cup fn(Q)) = \emptyset$ then there exists a process $Q^{'}$ such that $Q \xrightarrow{\alpha}_L Q^{'}$ and $P^{'}\mathbf{S}Q^{'}$

P and Q are *late bisimilar* written $P \dot{\sim}_L Q$ if there exists a strong late bisimulation $\mathbf{S}$ such that $P\mathbf{S}Q$.

Example Strong late bisimulation is not closed under substitution in general:

$$a(u).0|\bar{b}v.0 \ \dot{\sim}_L \ a(u).\bar{b}v.0 + \bar{b}v.a(u).0$$

and the bisimulation(without the symmetric part) is the following:

$$\{(a(u).0|\bar{b}v.0, a(u).\bar{b}v.0 + \bar{b}v.a(u).0), \ \ (a(u).0|0, a(u).0), \ \ (0|0, 0), \ \ (0|\bar{b}v.0, \bar{b}v.0)\}$$

If we apply the substitution $\{a/b\}$ to each process then they are not strongly bisimilar anymore because $(a(u).0|\bar{b}v.0)\{a/b\}$ is $a(u).0|\bar{a}v.0$ and this process can perform an invisible action whether $(a(u).\bar{b}v.0 + \bar{b}v.a(u).0)\{a/b\}$ cannot.

We refer to strong late bisimulation as strong *ground* late bisimulation, because it is not preserved by substitution.

Proposition 2.6.1. If $P \dot{\sim} Q$ and σ is injective then $P\sigma \dot{\sim} Q\sigma$

Proposition 2.6.2. $\dot{\sim}_L$ is an equivalence

Proposition 2.6.3. $\dot{\sim}_L$ is preserved by all operators except input prefix

Definition 2.6.2. Two processes P and Q are *strong late equivalent* written $P \sim_L Q$ is for each substitution σ $P\sigma \dot{\sim}_L Q\sigma$

Example If $z \notin fn(R) \cup \{x\}$ then $x(y).R \dot{\sim}_L (z)x(y).R$

2.6.2 Early bisimilarity

Definition 2.6.3. A *strong early bisimulation*(according to [4])is a symmetric binary relation $\mathbf{S}$ on processes such that for each process P and Q: $P\mathbf{S}Q$, $P \xrightarrow{\alpha}_E P^{'}$ and $bn(\alpha) \cap (fn(P) \cup fn(Q)) = \emptyset$ implies that there exists $Q^{'}$ such that $Q \xrightarrow{\alpha}_E Q^{'}$ and $P^{'}\mathbf{S}Q^{'}$. P and Q are *early bisimilar* written $P\dot{\sim}_E Q$ if there exists a strong early bisimulation $\mathbf{S}$ such that $P\mathbf{S}Q$

Definition 2.6.4. Two processes P and Q are *strong early equivalent* written $P \sim_E Q$ if for each substitution σ $P\sigma\dot{\sim}_E Q\sigma$

2.6.3 Congruence

Definition 2.6.5. We say that two agents P and Q are *strongly congruent*, written $P \sim Q$ if

$$P\sigma\dot{\sim}Q\sigma \text{ for all substitution } \sigma$$

Proposition 2.6.4. Strong congruence is the largest congruence in bisimilarity.

2.6.4 Open bisimilarity

Definition 2.6.6. A *distinction* is a finite symmetric and irreflexive binary relation on names. A substitution σ *respects* a distinction D if for each name a, b aDb implies $\sigma(a) \neq \sigma(b)$. We write $D\sigma$ for the composition of the two relation.

Definition 2.6.7. An *strong open simulation*(according to [4]) is $\{S_D\}_{D\in\mathbb{D}}$ a family of binary relations on processes such that for each process P, Q, for each distinction $D \in \mathbb{D}$, for each name substitution σ which respects D if PS_DQ, $P\sigma \xrightarrow{\alpha} P^{'}$ and $bn(\alpha) \cap (fn(P\sigma) \cup fn(Q\sigma)) = \emptyset$ then:

- if $\alpha = \overline{a}(x)$ then there exists $Q^{'}$ such that $Q\sigma \xrightarrow{\overline{a}(x)} Q^{'}$ and $P^{'}S_{D^{'}}Q^{'}$ where $D^{'} = D\sigma \cup \{x\} \times (fn(P\sigma) \cup fn(Q\sigma)) \cup (fn(P\sigma) \cup fn(Q\sigma)) \times \{x\}$
- if α is not a bound output then there exists $Q^{'}$ such that $Q\sigma \xrightarrow{\alpha} Q^{'}$ and $P^{'}S_{D\sigma}Q^{'}$

P and Q are *open D bisimilar*, written $P\dot{\sim}_O^D Q$ if there exists a member S_D of an open bisimulation such that PS_DQ; they are *open bisimilar* if they are open $\emptyset$ bisimilar, written $P\dot{\sim}_O D$.

Chapter 3

Multi π calculus with strong output

3.1 Syntax

As we did with π calculus, we suppose that we have a countable set of names $\mathbb{N}$, ranged over by lower case letters $a, b, \cdots, z$. This names are used for communication channels and values. Furthermore we have a set of identifiers, ranged over by A. We represent the agents or processes by upper case letters $P, Q, \cdots$. A multi π process, in addiction to the same actions of a π process, can perform also a strong prefix output:

$$\pi ::= \overline{x}y \mid x(z) \mid \underline{\overline{x}y} \mid \tau$$

The process are defined, just as original π calculus, by the following grammar:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P + Q \mid (\nu x)P \mid A(y_1, \cdots, y_n)$$

and they have the same intuitive meaning as for the π calculus. The strong prefix output allows a process to make an atomic sequence of actions, so that more than one process can synchronize on this sequence. For the moment we allow the strong prefix to be on output names only. Also one can use the strong prefix only as an action prefixing for processes that can make at least a further action.

Multi π calculus is a conservative extension of the π calculus in the sense that: any π calculus process p is also a multi π calculus process and the semantic of p according to the SOS rules of π calculus is the same as the semantic of p according to the SOS rules of multi π calculus.

We have to extend the following definition to deal with the strong prefix:

$$B(\underline{\overline{x}y}.Q, I) = B(Q, I) \quad F(\underline{\overline{x}y}.Q, I) = \{x, \overline{x}, y, \overline{y}\} \cup F(Q, I)$$

3.2 Operational semantic

3.2.1 Early operational semantic with structural congruence

The semantic of a multi π process is labeled transition system such that

- the nodes are multi π calculus process. The set of node is $\mathbb{P}_m$
- the actions are multi π calculus actions. The set of actions is $\mathbb{A}_m$, we use $\alpha, \alpha_1, \alpha_2, \cdots$ to range over the set of actions, we use $\sigma, \sigma_1, \sigma_2, \cdots$ to range over the set $\mathbb{A}_m^+ \cup \{\tau\}$. Note that σ is a non empty sequence of actions.
- the transition relations is $\rightarrow \subseteq \mathbb{P}_m \times (\mathbb{A}_m^+ \cup \{\tau\}) \times \mathbb{P}_m$

In this case, a label can be a sequence of prefixes, whether in the original π calculus a label can be only a prefix. We use the symbol $\cdot$ to denote the concatenation operator.

Definition 3.2.1. The *early transition relation* is the smallest relation induced by the rules in table 3.1.

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{SOutSeq}\ \frac{P \xrightarrow{\sigma} Q \qquad |\sigma| > 1}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\sigma} Q} \qquad \textbf{SOut}\ \frac{P \xrightarrow{\alpha} Q \qquad \alpha\ output}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\alpha} Q} \qquad \textbf{SOutTau}\ \frac{P \xrightarrow{\tau} Q}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y} Q}$$

$$\textbf{EComSng}\ \frac{P \xrightarrow{\overline{x}y} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad \textbf{EComSeq}\ \frac{P \xrightarrow{\overline{x}y\cdot\sigma} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\sigma} P'|Q'}$$

$$\textbf{Cong}\ \frac{P \equiv P' \qquad P' \xrightarrow{\sigma} Q}{P \xrightarrow{\sigma} Q} \qquad \textbf{Par}\ \frac{P \xrightarrow{\sigma} P'}{P|Q \xrightarrow{\sigma} P'|Q}$$

$$\textbf{Sum}\ \frac{P \xrightarrow{\sigma} P'}{P + Q \xrightarrow{\sigma} P'} \qquad \textbf{Res}\ \frac{P \xrightarrow{\sigma} P' \qquad z \notin n(\alpha)}{(\nu)zP \xrightarrow{\sigma} (\nu)zP'}$$

Table 3.1: Multi π early semantic with structural congruence

Lemma 3.2.1. If $P \xrightarrow{\sigma} Q$ then only one of the following cases hold:

- $|\sigma| = 1$
- $|\sigma| > 1$ and all the actions are output.

Example Multi-party synchronization. We show an example of a derivation of three processes that synchronize.

$$\textbf{Res}\ \frac{x \notin n(\tau) \qquad \textbf{EComSeq}\ \dfrac{\underline{\overline{x}y}.\overline{x}y.0|x(y).0 \xrightarrow{\overline{x}y} 0|0 \qquad \textbf{Inp}\ \dfrac{}{x(y).0 \xrightarrow{xy} 0}}{((\underline{\overline{x}y}.\overline{x}y.0|x(y).0)|x(y).0) \xrightarrow{\tau} ((0|0)|0)}}{(\nu x)((\underline{\overline{x}y}.\overline{x}y.0|x(y).0)|x(y).0) \xrightarrow{\tau} (\nu x)((0|0)|0)}$$

$$\textbf{EComSng}\ \frac{\textbf{SOut}\ \dfrac{\textbf{Out}\ \dfrac{}{\overline{x}y.0 \xrightarrow{\overline{x}y} 0}}{\underline{\overline{x}y}.\overline{x}y.0 \xrightarrow{\overline{x}y\cdot\overline{x}y} 0} \qquad x(y).0 \xrightarrow{xy} 0}{\underline{\overline{x}y}.\overline{x}y.0|x(y).0 \xrightarrow{\overline{x}y} 0|0}$$

Example Transactional synchronization In this setting two process cannot synchronize on a sequence of actions with length greater than one. This is because of the rules *EComSng* and *EComSeq*.

3.2.2 Low level semantic

This section contains the definition of an alternative semantic for multi π. First we define a low level version of the multi π calculus(here with strong prefixing on output only), we call this language low multi π. The low multi π is the multi π enriched with a marked or intermediate process $*P$:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P + Q \mid (\nu x)P \mid A \mid *P$$

$$\pi ::= \overline{x}y \mid x(y) \mid \underline{\overline{x}y} \mid \tau$$

Definition 3.2.2. The low level transition relation is the smallest relation induced by the rules in table 3.2 in which P stands for a process without mark, L stands for a process with mark and S can stand for both.

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{SOutLow}\ \frac{}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y} *P} \qquad \textbf{StarEps}\ \frac{S \xrightarrow{\epsilon} S^{'}}{*S \xrightarrow{\epsilon} S^{'}} \qquad \textbf{StarOut}\ \frac{S \xrightarrow{\overline{x}y} S^{'}}{*S \xrightarrow{\overline{x}y} S^{'}}$$

$$\textbf{Com1}\ \frac{P \xrightarrow{\overline{x}y} P^{'} \qquad Q \xrightarrow{xy} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}|Q^{'}}$$

$$\textbf{Com2L}\ \frac{L_1 \xrightarrow{\overline{x}y} L_1^{'} \qquad P \xrightarrow{xy} Q}{L_1|P \xrightarrow{\epsilon} L_1^{'}|Q} \qquad \textbf{Com2R}\ \frac{P \xrightarrow{xy} Q \qquad L_1 \xrightarrow{\overline{x}y} L_1^{'}}{P|L_1 \xrightarrow{\epsilon} Q|L_1^{'}}$$

$$\textbf{Com3L}\ \frac{P \xrightarrow{\overline{x}y} L \qquad Q \xrightarrow{xy} Q^{'}}{P|Q \xrightarrow{\epsilon} L|Q^{'}} \qquad \textbf{Com3R}\ \frac{P \xrightarrow{xy} P^{'} \qquad Q \xrightarrow{\overline{x}y} L}{P|Q \xrightarrow{\epsilon} P^{'}|L}$$

$$\textbf{Com4L}\ \frac{L \xrightarrow{\overline{x}y} Q \qquad P \xrightarrow{xy} P^{'}}{L|P \xrightarrow{\tau} Q|P^{'}} \qquad \textbf{Com4R}\ \frac{P \xrightarrow{xy} P^{'} \qquad L \xrightarrow{\overline{x}y} Q}{P|L \xrightarrow{\tau} P^{'}|Q}$$

$$\textbf{Res}\ \frac{S \xrightarrow{\gamma} S^{'} \qquad y \notin n(\gamma)}{(\nu y)S \xrightarrow{\gamma} (\nu y)S^{'}} \qquad \textbf{Sum}\ \frac{P \xrightarrow{\gamma} S}{P+Q \xrightarrow{\gamma} S} \qquad \textbf{Cong}\ \frac{P \equiv P^{'} \qquad P^{'} \xrightarrow{\gamma} S}{P \xrightarrow{\gamma} S}$$

$$\textbf{Par1L}\ \frac{S \xrightarrow{\gamma} S^{'}}{S|Q \xrightarrow{\gamma} S^{'}|Q} \qquad \textbf{Par1R}\ \frac{S \xrightarrow{\gamma} S^{'}}{Q|S \xrightarrow{\gamma} Q|S^{'}}$$

Table 3.2: Low multi π early semantic with structural congruence

Lemma 3.2.2. For all unmarked processes P, Q and marked processes L, L_1, L_2.

- if $L_1 \xmapsto{\alpha} L_2$ or $P \xmapsto{\alpha} L$ then α can only be an output or an ϵ
- if $L \xmapsto{\alpha} P$ then α can only be an output or a τ
- if $P \xmapsto{\alpha} Q$ then α cannot be an ϵ

Definition 3.2.3. Let P, Q be unmarked processes and $L_1, \cdots, L_{k-1}$ marked processes. We define the derivation relation $\rightarrow_s$ in the following way:

$$\textbf{Low}\ \frac{P \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} Q \qquad k \geq 1}{P \xrightarrow{\gamma_1 \cdots \gamma_k}_s Q}$$

We need to be precise about the concatenation operator $\cdot$ since we have introduced the new label ϵ. Let a be an action such that $a \neq \tau$ and $a \neq \epsilon$ then the following rules hold:

$$\begin{array}{lll} \epsilon \cdot a = a \cdot \epsilon = a & \epsilon \cdot \epsilon = \epsilon & \tau \cdot \epsilon = \epsilon \cdot \tau = \tau \\ \tau \cdot a = a \cdot \tau = a & \tau \cdot \tau = \tau & \end{array}$$

Example Multi-parti synchronization

$$\textbf{Par1L}\ \cfrac{\textbf{Par1L}\ \cfrac{\textbf{Com3L}\ \cfrac{\textbf{SOutLow}\ \cfrac{}{\underline{\overline{x}a}.\overline{x}b.\overline{x}c.P \xmapsto{\overline{x}a} *\underline{\overline{x}b}.\overline{x}c.P} \qquad \textbf{Inp}\ \cfrac{}{x(d).Q \xrightarrow{xa} Q\{a/d\}}}{\underline{\overline{x}a}.\underline{\overline{x}b}.\overline{x}c.P|x(d).Q \xmapsto{\epsilon} *\underline{\overline{x}b}.\overline{x}c.P|Q\{a/d\}}}{\underline{\overline{x}a}.\underline{\overline{x}b}.\overline{x}c.P|x(d).Q|x(e).R \xmapsto{\epsilon} *\underline{\overline{x}b}.\overline{x}c.P|Q\{a/d\}|x(e).R}}{\underline{\overline{x}a}.\underline{\overline{x}b}.\overline{x}c.P|x(d).Q|x(e).R|x(f).S \xmapsto{\epsilon} *\underline{\overline{x}b}.\overline{x}c.P|Q\{a/d\}|x(e).R|x(f).S}$$

$$\textbf{Par1L}\ \cfrac{\textbf{Com2L}\ \cfrac{\textbf{Par1L}\ \cfrac{\textbf{StarOut}\ \cfrac{\textbf{SOutLow}\ \cfrac{}{\underline{\overline{x}b}.\overline{x}c.P \xmapsto{\overline{x}b} *\overline{x}c.P}}{*\underline{\overline{x}b}.\overline{x}c.P \xmapsto{\overline{x}b} *\overline{x}c.P}}{*\underline{\overline{x}b}.\overline{x}c.P|Q\{a/d\} \xmapsto{\overline{x}b} *\overline{x}c.P|Q\{a/d\}} \qquad \textbf{EInp}\ \cfrac{}{x(e).R \xmapsto{xb} R\{b/e\}}}{*\underline{\overline{x}b}.\overline{x}c.P|Q\{a/d\}|x(e).R \xmapsto{\epsilon} *\overline{x}c.P|Q\{a/d\}|R\{b/e\}}}{*\underline{\overline{x}b}.\overline{x}c.P|Q\{a/d\}|x(e).R|x(f).S \xmapsto{\epsilon} *\overline{x}c.P|Q\{a/d\}|R\{b/e\}|x(f).S}$$

$$\textbf{Com4L}\ \cfrac{\textbf{Par1L}\ \cfrac{\textbf{Par1L}\ \cfrac{\textbf{StarOut}\ \cfrac{\textbf{Out}\ \cfrac{}{\overline{x}c.P \xmapsto{\overline{x}c} P}}{*\overline{x}c.P \xmapsto{\overline{x}c} P}}{*\overline{x}c.P|Q\{a/d\} \xmapsto{\overline{x}c} P|Q\{a/d\}}}{*\overline{x}c.P|Q\{a/d\}|R\{b/e\} \xmapsto{\overline{x}c} P|Q\{a/d\}|R\{b/e\}} \qquad \textbf{EInp}\ \cfrac{}{x(f).S \xmapsto{xc} R\{c/f\}}}{*\overline{x}c.P|Q\{a/d\}|R\{b/e\}|x(f).S \xmapsto{\tau} P|Q\{a/d\}|R\{b/e\}|S\{c/f\}}$$

Proposition 3.2.3. Let $\rightarrow$ be the relation defined in table 3.1. If $P \xrightarrow{\sigma} Q$ then there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

Proof. The proof is by induction on the depth of the derivation tree of $P \xrightarrow{\sigma} Q$ and by cases on the last rule used in the derivation:

EInp, Out, Tau These rules are also in table 3.2 so we can derive $P \xmapsto{\sigma} Q$.

SOutSeq : the last part of the derivation tree looks like this:

$$\textbf{SOutSeq}\ \frac{P_1 \xrightarrow{\sigma} Q \qquad |\sigma| > 1}{\underline{\overline{x}y}.P_1 \xrightarrow{\overline{x}y \cdot \sigma} Q}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

then a proof of the conclusion follows from:

$$\textbf{SOutLow} \, \frac{}{\underline{\overline{x}y}.P_1 \overset{\overline{x}y}{\longmapsto} *P_1} \qquad \textbf{Star} \, \frac{P_1 \overset{\gamma_1}{\longmapsto} L_1}{*P_1 \overset{\gamma_1}{\longmapsto} L_1}$$

$SOut$: this case is similar to the previous.

$SOutTau$: this case is similar to the previous observing that $\overline{x}y \cdot \tau = \overline{x}y$.

Sum : the last part of the derivation tree looks like this:

$$\textbf{Sum} \, \frac{P_1 \xrightarrow{\sigma} Q}{P_1 + P_2 \xrightarrow{\sigma} Q}$$

for the inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

A proof of the conclusion is:

$$\textbf{Sum} \, \frac{P_1 \overset{\gamma_1}{\longmapsto} L_1}{P_1 + P_2 \overset{\gamma_1}{\longmapsto} L_1}$$

$Cong$: this case is similar to the previous.

$EComSng$: the last part of the derivation tree looks like this:

$$\textbf{Com} \, \frac{P_1 \xrightarrow{\overline{x}y} P_1^{'} \qquad Q_1 \xrightarrow{xy} Q_1^{'}}{P_1|Q_1 \xrightarrow{\tau} P_1^{'}|Q_1^{'}}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} P_1^{'} \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \overline{x}y$$

and there exist $R_1, \cdots, R_h$ and $\delta_1, \cdots, \delta_{h+1}$ with $h \geq 0$ such that

$$Q_1 \overset{\delta_1}{\longmapsto} R_1 \overset{\delta_2}{\longmapsto} R_2 \cdots R_{h-1} \overset{\delta_h}{\longmapsto} R_h \overset{\delta_{h+1}}{\longmapsto} Q_1^{'} \quad \text{and} \quad \delta_1 \cdot \ldots \cdot \delta_{h+1} = xy$$

For lemma 3.2.2 there cannot be an input action in a transition involving marked processes so h must be 0 and $Q_1 \overset{\delta_1}{\longmapsto} Q_1^{'}$ with $\delta_1 = xy$. Just one of the γs is $\overline{x}y$ and the others are ϵ or τ. We can have three different cases now:

$\gamma_1 = \overline{x}y$: A proof of the conclusion is:

$$P_1|Q_1 \overset{\tau}{\longmapsto} L_1|Q_1^{'} \overset{\epsilon}{\longmapsto} L_2|Q_1^{'} \cdots \overset{\epsilon}{\longmapsto} L_k|Q_1^{'} \overset{\tau}{\longmapsto} P_1^{'}|Q_1^{'}$$

we derive the first transition with rule $Com3L$, whether for the other transition we use the rule $Par1L$.

$\gamma_i = \overline{x}y$: A proof of the conclusion is:

$$P_1|Q_1 \overset{\epsilon}{\longmapsto} L_1|Q_1 \cdots \overset{\epsilon}{\longmapsto} L_{i-1}|Q_1 \overset{\epsilon}{\longmapsto} L_i|Q_1^{'} \overset{\epsilon}{\longmapsto} L_{i+1}|Q_1^{'} \cdots \overset{\epsilon}{\longmapsto} L_k|Q_1^{'} \overset{\tau}{\longmapsto} P_1^{'}|Q_1^{'}$$

we derive the transaction $L_{i-1}|Q_1 \overset{\epsilon}{\longmapsto} L_i|Q_1^{'}$ with rule $Com2L$, whether for the other transactions we use the rule $Par1L$.

$\gamma_{k+1} = \overline{x}y$ similar.

Res : the last part of the derivation tree looks like this:

$$\textbf{Res}\ \frac{P_1 \ \xrightarrow{\sigma}\ Q_1 \qquad z \notin n(\sigma)}{(\nu z)P_1 \ \xrightarrow{\sigma}\ (\nu z)Q_1}$$

for the inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q_1 \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

We can apply the rule *Res* to each of the previous transitions because

$$z \notin n(\sigma) \text{ implies } z \notin n(\gamma_i) \text{ for each } i$$

and then get a proof of the conclusion:

$$(\nu z)P_1 \xmapsto{\gamma_1} (\nu z)L_1 \xmapsto{\gamma_2} (\nu z)L_2 \cdots (\nu z)L_{k-1} \xmapsto{\gamma_k} (\nu z)L_k \xmapsto{\gamma_{k+1}} (\nu z)Q_1$$

Par : this case is similar to the previous.

EComSeq : the last part of the derivation tree looks like this:

$$\textbf{EComSeq}\ \frac{P_1 \xrightarrow{\overline{x}y \cdot \sigma} P_1^{'} \qquad Q_1 \xrightarrow{xy} Q_1^{'}}{P_1|Q_1 \xrightarrow{\sigma} P_1^{'}|Q_1^{'}}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} P_1^{'} \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \overline{x}y \cdot \sigma$$

For inductive hypothesis and lemma 3.2.2 $Q_1 \xmapsto{xy} Q_1^{'}$. We can have two different cases now depending on where the first $\overline{x}y$ is:

$\gamma_1 = \overline{x}y$: A proof of the conclusion is:

$$P_1|Q_1 \xmapsto{\epsilon} L_1|Q_1^{'} \xmapsto{\gamma_2} L_2|Q_1^{'} \cdots \xmapsto{\gamma_k} L_k|Q_1^{'} \xmapsto{\gamma_{k+1}} P_1^{'}|Q_1^{'}$$

we derive the first transition with rule *Com*3*L*, whether for the other transactions we use the rule *Par*1*L*. Since $\gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \overline{x}y \cdot \sigma$ and $\gamma_1 = \overline{x}y$ then $\epsilon \cdot \gamma_2 \cdot \ldots \cdot \gamma_{k+1} = \sigma$

$\gamma_i = \overline{x}y$: A proof of the conclusion is:

$$P_1|Q_1 \xmapsto{\epsilon} L_1|Q_1 \cdots \xmapsto{\epsilon} L_{i-1}|Q_1 \xmapsto{\epsilon} L_i|Q_1^{'} \xmapsto{\gamma_{i+1}} L_{i+1}|Q_1^{'} \cdots \xmapsto{\gamma_k} L_k|Q_1^{'} \xmapsto{\gamma_{k+1}} P_1^{'}|Q_1^{'}$$

we derive the transition $L_{i-1}|Q_1 \xmapsto{\epsilon} L_i|Q_1^{'}$ with rule *Com*2*L*, whether for the other transactions of the premises we use the rule *Par*1*L*.

$\gamma_{k+1} = \overline{x}y$: cannot happen because σ is not empty.

□

We would like to prove the converse of the previous proposition, namely: if there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

then $P \xrightarrow{\sigma} Q$. But this is false as shown by these examples:

Example Interleaving

$$\textbf{Par1L}\ \dfrac{\textbf{Com3L}\ \dfrac{\textbf{SOutLow}\ \dfrac{}{\underline{\overline{x}y}.\underline{\overline{a}b}.\overline{x}y.0 \xmapsto{\overline{x}y} *\underline{\overline{a}b}.\overline{x}y.0} \qquad \textbf{EInp}\ \dfrac{}{x(y).0 \xmapsto{xy} 0}}{\underline{\overline{x}y}.\underline{\overline{a}b}.\overline{x}y.0|x(y).0 \xmapsto{\epsilon} *\underline{\overline{a}b}.\overline{x}y.0|0}}{\underline{\overline{x}y}.\underline{\overline{a}b}.\overline{x}y.0|x(y).0|x(y).0 \xmapsto{\epsilon} *\underline{\overline{a}b}.\overline{x}y.0|0|x(y).0}$$

$$\textbf{Par1L}\ \dfrac{\textbf{Par1L}\ \dfrac{\textbf{StarOut}\ \dfrac{\textbf{SOutLow}\ \dfrac{}{\underline{\overline{a}b}.\overline{x}y.0 \xmapsto{\overline{a}b} *\overline{x}y.0}}{*\underline{\overline{a}b}.\overline{x}y.0 \xmapsto{\overline{a}b} *\overline{x}y.0}}{*\underline{\overline{a}b}.\overline{x}y.0|0 \xmapsto{\overline{a}b} *\overline{x}y.0|0}}{*\underline{\overline{a}b}.\overline{x}y.0|0|x(y).0 \xmapsto{\overline{a}b} *\overline{x}y.0|0|x(y).0}$$

$$\textbf{Com4L}\ \dfrac{\textbf{Par1L}\ \dfrac{\textbf{StarOut}\ \dfrac{\textbf{Out}\ \dfrac{}{\overline{x}y.0 \xmapsto{\overline{x}y} 0}}{*\overline{x}y.0 \xmapsto{\overline{x}y} 0}}{*\overline{x}y.0|0 \xmapsto{\overline{x}y} 0|0} \qquad \textbf{EInp}\ \dfrac{}{x(y).0 \xmapsto{xy} 0}}{*\overline{x}y.0|0|x(y).0 \xmapsto{\tau} 0|0|0}$$

this prove:

$$\underline{\overline{x}y}.\underline{\overline{a}b}.\overline{x}y.0|x(y).0|x(y).0 \xmapsto{\epsilon} *\underline{\overline{a}b}.\overline{x}y.0|0|x(y).0 \xmapsto{\overline{a}b} *\overline{x}y.0|0|x(y).0 \xmapsto{\tau} 0|0|0$$

but there is no way to prove

$$\underline{\overline{x}y}.\underline{\overline{a}b}.\overline{x}y.0|x(y).0|x(y).0 \xrightarrow{\overline{a}b} 0|0|0$$

Example Transactional synchronization

$$\textbf{Com3L}\ \dfrac{\textbf{SOutLow}\ \dfrac{}{\underline{\overline{x}y}.\overline{x}y.0 \xmapsto{\overline{x}y} *\overline{x}y.0} \qquad \textbf{EInp}\ \dfrac{}{x(y).x(y).0 \xmapsto{xy} x(y).0}}{\underline{\overline{x}y}.\overline{x}y.0|x(y).x(y).0 \xmapsto{\epsilon} *\overline{x}y.0|x(y).0}$$

$$\textbf{Com4L}\ \dfrac{\textbf{StarOut}\ \dfrac{\textbf{Out}\ \dfrac{}{\overline{x}y.0 \xmapsto{\overline{x}y} 0}}{*\overline{x}y.0 \xmapsto{\overline{x}y} 0} \qquad \textbf{EInp}\ \dfrac{}{x(y).0 \xmapsto{xy} 0}}{*\overline{x}y.0|x(y).0 \xmapsto{\tau} 0|0}$$

this prove:

$$\underline{\overline{x}y}.\overline{x}y.0|x(y).x(y).0 \xmapsto{\epsilon} *\overline{x}y.0|x(y).0 \xmapsto{\tau} 0|0$$

but we cannot derive

$$\underline{\overline{x}y}.\overline{x}y.0|x(y).x(y).0 \xrightarrow{\tau} 0|0$$

also we do not want to derive this transaction because the second process does not start with a strong prefix.

There is a much weaker propositions we can prove:

Proposition 3.2.4. Let $\rightarrow$ be the relation defined in table 3.1. Let α be an action. If $P \xmapsto{\alpha} Q$ then $P \xrightarrow{\alpha} Q$.

Proof. The proof is by induction the depth of the derivation of $P \xmapsto{\alpha} Q$ and by cases on the last rule used in the derivation:

$Out, EInp, Tau$ These rules are also in table 4.1 so we can derive $P \xrightarrow{\alpha} Q$.

$Com1$:

$$\textbf{Com1}\ \frac{P_1 \overset{\overline{x}y}{\longmapsto} Q_1 \qquad P_2 \overset{xy}{\longmapsto} Q_2}{P_1|P_2 \overset{\tau}{\longmapsto} Q_1|Q_2}$$

for inductive hypothesis $P_1 \xrightarrow{\overline{x}y} Q_1$ and $P_2 \xrightarrow{xy} Q_2$ so for rule Com $P_1|P_2 \xrightarrow{\tau} Q_1|Q_2$

Sum :

$$\textbf{Sum}\ \frac{P_1 \overset{\alpha}{\longmapsto} Q}{P_1 + P_2 \overset{\alpha}{\longmapsto} Q}$$

for inductive hypothesis $P_1 \xrightarrow{\alpha} Q$ and for rule Sum $P_1 + P_2 \xrightarrow{\alpha} Q$.

Res the first transition is:

$$\textbf{Res}\ \frac{P_1 \overset{\alpha}{\longmapsto} Q_1 \qquad z \notin n(\gamma_1)}{(\nu z)P_1 \overset{\alpha}{\longmapsto} (\nu z)Q_1}$$

for inductive hypothesis $P_1 \xrightarrow{\alpha} Q_1$ and for rule Res $(\nu z)P_1 \xrightarrow{\alpha} (\nu z)Q_1$.

others : other cases are similar.

□

Since it's important to give a low level semantic which is equivalent to the high level one, we can propose a change to the low level semantic that gets closer to our purpose. We replace the rule $Com3L$, $Com3R$, $Com2L$ and $Com2R$ with:

$$\textbf{Com2LStop}\ \frac{L_1 \overset{\overline{x}y}{\longmapsto} L_2 \qquad P \overset{xy}{\longmapsto} Q}{L_1|P \overset{\epsilon}{\longmapsto} L_2|stop(Q)} \qquad \textbf{Com2RStop}\ \frac{P \overset{xy}{\longmapsto} Q \qquad L_1 \overset{\overline{x}y}{\longmapsto} L_2}{P|L_1 \overset{\epsilon}{\longmapsto} stop(Q)|L_2}$$

$$\textbf{Com3LStop}\ \frac{P \overset{\overline{x}y}{\longmapsto} L \qquad Q \overset{xy}{\longmapsto} Q^{'}}{P|Q \overset{\epsilon}{\longmapsto} L|stop(Q^{'})} \qquad \textbf{Com3RStop}\ \frac{P \overset{xy}{\longmapsto} P^{'} \qquad Q \overset{\overline{x}y}{\longmapsto} L}{P|Q \overset{\epsilon}{\longmapsto} stop(P^{'})|L}$$

where $stop(P)$ is a multi π process which cannot make any transition.

Definition 3.2.4. The *erase function er* is a function that eliminates the *stop* mark on processes. Its definition is straightforward.

Proposition 3.2.5. Let $\rightarrow$ be the relation defined in table 3.1. If $P \xrightarrow{\sigma} Q$ then there exist $L_1, \cdots, L_k$ with $k \geq 0$ such that

$$P \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q^{'} \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma \quad \text{and} \quad er(Q^{'}) = Q$$

Proof. The proof of the first part of this proposition is almost exactly as the proof of proposition 3.2.3. □

Proposition 3.2.6. Let $\rightarrow$ be the relation defined in table 3.1. If there exist L_1, $\cdots$, L_k with $k \geq 1$ such that

$$P \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q$$

and

- if all γs are ϵ or τ then $P \xrightarrow{\tau} er(Q)$
- if there is an output $\overline{x}y$ in the γs and all the other all γs are ϵ or τ then $P \xrightarrow{\overline{x}y} er(Q)$

Proof. The proof is by induction on the depth of the derivation of $P \overset{\gamma_1}{\longmapsto} L_1$ and by cases on the last rule used in the derivation:

SOutLow :

$$\textbf{SOutLow}\ \frac{}{\underbrace{\underline{\overline{x}y}.P_1}_{P} \overset{\overline{x}y}{\longmapsto} \underbrace{*P_1}_{L_1}}$$

since $*P_1$ has a mark at the top level, the last rule used to derive $*P_1 \overset{\gamma_2}{\longmapsto}$ has to be $StarEps$ so we have $P_1 \overset{\gamma_2}{\longmapsto} L_2$ or $P_1 \overset{\gamma_2}{\longmapsto} Q$ depending on k. We can build the following chain of transition:

$$P_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q$$

since γ_1 is an output, the other γs are ϵ or τ, then we can apply the inductive hypothesis to get $P_1 \xrightarrow{\tau} er(Q)$. Now a proof of the conclusion is

$$\textbf{SOutTau}\ \frac{P_1 \xrightarrow{\tau} er(Q)}{\underline{\overline{x}y}.P_1 \xrightarrow{\overline{x}y} er(Q)}$$

Sum the first transition is:

$$\textbf{Sum}\ \frac{P_1 \overset{\gamma_1}{\longmapsto} L_1}{P_1 + P_2 \overset{\gamma_1}{\longmapsto} L_1}$$

so we can build the following chain of transition:

$$P_1 \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q$$

apply the inductive hypothesis to get $P_1 \xrightarrow{\alpha} er(Q)$ where α is τ or an output. Now a proof of the conclusion is

$$\textbf{Sum}\ \frac{P_1 \xrightarrow{\alpha} er(Q)}{P_1 + P_2 \xrightarrow{\alpha} er(Q)}$$

Res the first transition is:

$$\textbf{Res}\ \frac{P_1 \overset{\gamma_1}{\longmapsto} L_1^{'} \qquad z \notin n(\gamma_1)}{(\nu z)P_1 \overset{\gamma_1}{\longmapsto} (\nu z)L_1^{'}}$$

given that L_1 has a restriction at the top level, all the other intermediate processes $L_2, \cdots, L_k$ and Q have the same restriction at the top level. This is because the only rule whose conclusion is a transition that start from a possibly marked process with a restriction at its top level is *Res*. So the last rule used to prove all transition is *Res*.

$$\textbf{Res}\ \frac{L_k^{'} \overset{\gamma_{k+1}}{\longmapsto} Q^{'} \qquad z \notin n(\tau)}{(\nu z)L_k^{'} \overset{\gamma_{k+1}}{\longmapsto} (\nu z)Q^{'}} \qquad \textbf{Res}\ \frac{L_i^{'} \overset{\gamma_i}{\longmapsto} L_{i+1}^{'} \qquad z \notin n(\epsilon)}{(\nu z)L_i^{'} \overset{\gamma_i}{\longmapsto} (\nu z)L_{i+1}^{'}}$$

we can build the following chain of transitions:

$$P_1 \overset{\gamma_1}{\longmapsto} L_1^{'} \overset{\gamma_2}{\longmapsto} L_2^{'} \cdots L_{k-1}^{'} \overset{\gamma_k}{\longmapsto} L_k^{'} \overset{\gamma_{k+1}}{\longmapsto} Q^{'}$$

then apply the inductive hypothesis to get $P_1 \xrightarrow{\alpha} er(Q^{'})$. A proof of the conclusion can be

$$\textbf{Res}\ \frac{P_1 \xrightarrow{\alpha} er(Q^{'}) \qquad z \notin n(\tau)}{(\nu z)P_1 \xrightarrow{\alpha} (\nu z)er(Q^{'}) = er((\nu z)Q^{'})}$$

$Cong$ the last rule of the derivation of the first transition is:

$$\textbf{Cong}\ \frac{P^{'} \equiv P \qquad \xmapsto{\gamma_1} L_1}{P \xmapsto{\gamma_1} L_1}$$

We derive the following chain of transition:

$$P^{'} \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q$$

for inductive hypothesis $P^{'} \xrightarrow{\alpha} er(Q)$. A proof of the conclusion is

$$\textbf{Cong}\ \frac{P^{'} \equiv P \qquad P^{'} \xrightarrow{\alpha} er(Q)}{P \xrightarrow{\alpha} er(Q)}$$

$Com3LStop$: the last part of the derivation of the first transition is:

$$\textbf{Com3LStop}\ \frac{P_1 \xmapsto{\overline{x}y} L_1^{'} \qquad P_2 \xmapsto{xy} Q_2}{P_1|P_2 \xmapsto{\epsilon} L_1^{'}|stop(Q_2)}$$

the derivations of all other transitions can end only with an instance of $Par1L$ so we have:

$$\textbf{Par1L}\ \frac{L_i^{'} \xmapsto{\gamma_i} L_{i+1}^{'}}{L_i^{'}|stop(Q_2) \xmapsto{\gamma_i} L_{i+1}^{'}|stop(Q_2)} \qquad \textbf{Par1L}\ \frac{L_k^{'} \xmapsto{\gamma_{k+1}} Q_1}{L_i^{'}|stop(Q_2) \xmapsto{\gamma_{k+1}} Q_1|stop(Q_2)}$$

We derive the following chain of transition:

$$P_1 \xmapsto{\overline{x}y} L_1^{'} \xmapsto{\gamma_2} L_2^{'} \cdots L_{k-1}^{'} \xmapsto{\gamma_k} L_k^{'} \xmapsto{\gamma_{k+1}} Q_1$$

for inductive hypothesis $P_1 \xrightarrow{\overline{x}y} er(Q_1)$. A proof of the conclusion is

$$\textbf{EComSeq}\ \frac{P_1 \xrightarrow{\overline{x}} er(Q_1) \qquad P_2 \xmapsto{xy} Q_2}{P_1|P_2 \xrightarrow{\tau} er(Q_1)|Q_2}$$

$Par1L(1)$: the last part of the derivation of the first transition is:

$$\textbf{Par1L}\ \frac{P_1 \xmapsto{\gamma_1} L_1^{'}}{P_1|P_2 \xmapsto{\gamma_1} L_1^{'}|P_2}$$

the derivations of all the other transitions end with an instance of $Par1L$. We derive the following chain of transition:

$$P_1 \xmapsto{\gamma_1} L_1^{'} \xmapsto{\gamma_2} L_2^{'} \cdots L_{k-1}^{'} \xmapsto{\gamma_k} L_k^{'} \xmapsto{\gamma_k} Q_1$$

for inductive hypothesis $P_1 \xrightarrow{\alpha} er(Q_1)$. A proof of the conclusion is

$$\textbf{Par}\ \frac{P_1 \xrightarrow{\alpha} er(Q_1)}{P_1|P_2 \xrightarrow{\alpha} er(Q_1)|P_2^{'}}$$

$Par1L(2)$: the last part of the derivation of the first transition is:

$$\textbf{Par1L}\ \frac{P_1 \xmapsto{\gamma_1} L_1^{'}}{P_1|P_2 \xmapsto{\gamma_1} L_1^{'}|P_2}$$

there is one derivation that ends with an instance of $Com2LStop$ and the derivations of all the other transitions end with an instance of $Par1L$. We present here the case when the second transition ends with a $Com2LStop$, the other cases are similar. So

$$\textbf{Com2LStop}\ \frac{L_2^{'} \xrightarrow{\overline{x}y} L_2^{'} \qquad P_2 \xrightarrow{xy} P_2^{'}}{L_2^{'}|P_2 \xrightarrow{\epsilon} L_2^{'}|stop(P_2^{'})}$$

We derive the following chain of transition:

$$P_1 \xmapsto{\epsilon} L_1^{'} \xmapsto{\overline{x}y} L_2^{'} \xmapsto{\epsilon} \cdots \xmapsto{\epsilon} L_{k-1}^{'} \xmapsto{\epsilon} L_k^{'} \xmapsto{\tau} Q_1$$

for inductive hypothesis $P_1 \xrightarrow{\overline{x}y} er(Q_1)$. A proof of the conclusion is

$$\textbf{EComSeq}\ \frac{P_1 \xrightarrow{\overline{x}y} er(Q_1) \qquad P_2 \xrightarrow{xy} P_2^{'}}{P_1|P_2 \xrightarrow{\tau} er(Q_1)|P_2^{'}}$$

$Par1L(3)$: the last part of the derivation of the first transition is:

$$\textbf{Par1L}\ \frac{P_1 \xmapsto{\gamma_1} L_1^{'}}{P_1|P_2 \xmapsto{\gamma_1} L_1^{'}|P_2}$$

the derivation of the last transition ends with an instance of $Com4L$ and the derivations of all the other transitions end with an instance of $Par1L$. We derive the following chain of transition:

$$P_1 \xmapsto{\epsilon} L_1^{'} \xmapsto{\epsilon} L_2^{'} \cdots L_{k-1}^{'} \xmapsto{\epsilon} L_k^{'} \xmapsto{\overline{x}y} Q_1$$

for inductive hypothesis $P_1 \xrightarrow{\overline{x}y} er(Q_1)$. A proof of the conclusion is

$$\textbf{EComSeq}\ \frac{P_1 \xrightarrow{\overline{x}y} er(Q_1) \qquad P_2 \xrightarrow{xy} P_2^{'}}{P_1|P_2 \xrightarrow{\tau} er(Q_1)|P_2^{'}}$$

$\square$

3.2.3 Early operational semantic without structural congruence

Definition 3.2.5. The *late transition relation without structural congruence* is the smallest relation induced by the rules in table 3.3 and in table 3.4.

Example Scope extrusion with strong prefixing(1). $x \notin fn(y(z).Q|a(b).R|y(z).S)$. The following is the desired transition:

$$(\nu x)(\underline{\overline{y}x}.\underline{\overline{a}b}.\underline{\overline{y}x}.\overline{a}b.P))|y(z).Q|a(b).R|y(z).S|a(b).T \xrightarrow{\tau} (\nu x)(P|Q\{x/z\}|S\{x/z\})|R|T$$

It is possible to infer this transition in the semantic with structural congruence. But without structural congruence and the following spoce extrusion rules

$$\textbf{Opn}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad y \in obj(\sigma) \qquad y \notin sbj(\sigma)}{(\nu y)P \xrightarrow{opn(\sigma,y)} P^{'}}$$

$$\textbf{Cls}\ \frac{P \xrightarrow{\overline{x}(z)} P^{'} \qquad Q \xrightarrow{xz} Q^{'}}{P|Q \xrightarrow{\tau} (\nu z)(P^{'}|Q^{'})} \qquad \textbf{ClsSeq}\ \frac{P \xrightarrow{\overline{x}(z)\cdot\sigma} P^{'} \qquad Q \xrightarrow{xz} Q^{'}}{P|Q \xrightarrow{\sigma} (\nu z)(P^{'}|Q^{'})}$$

$$\textbf{SOut}\ \frac{n \geq 0}{\underline{\overline{x_1}y_1}.\dots.\underline{\overline{x_n}y_n}.\overline{x}y.P \xrightarrow{\widetilde{\overline{x}y}\cdot\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(z).P \xrightarrow{xw} P\{w/z\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{EComSeq}\ \frac{P \xrightarrow{\overline{x}y\cdot\sigma} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\sigma} P'|Q'} \qquad \textbf{ECom}\ \frac{P \xrightarrow{\overline{x}y} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\textbf{ParL}\ \frac{P \xrightarrow{\sigma} P' \qquad bn(\sigma) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\sigma} P'|Q} \qquad \textbf{ParR}\ \frac{Q \xrightarrow{\sigma} Q' \qquad bn(\sigma) \cap fn(P) = \emptyset}{P|Q \xrightarrow{\sigma} P|Q'}$$

$$\textbf{Res}\ \frac{P \xrightarrow{\sigma} P' \qquad z \notin n(\sigma)}{(\nu z)P \xrightarrow{\sigma} (\nu z)P'} \qquad \textbf{Ide}\ \frac{A(\tilde{x}) \stackrel{def}{=} P \qquad P\{\tilde{y}/\tilde{x}\} \xrightarrow{\sigma} Q}{A(\tilde{y}) \xrightarrow{\sigma} Q}$$

$$\textbf{SumL}\ \frac{P \xrightarrow{\sigma} P'}{P + Q \xrightarrow{\sigma} P'} \qquad \textbf{SumR}\ \frac{Q \xrightarrow{\sigma} Q'}{P + Q \xrightarrow{\sigma} Q'}$$

$$\textbf{Alph}\ \frac{P \equiv_\alpha Q \qquad Q \xrightarrow{\sigma} P'}{P \xrightarrow{\sigma} P'}$$

Table 3.3: Multiπ early semantic without structural congruence part 1

$$\textbf{Opn}\ \frac{P \xrightarrow{\sigma} P' \qquad y \in obj(\sigma) \qquad y \notin sbj(\sigma)}{(\nu y)P \xrightarrow{opn(\sigma,y)} P'} \qquad \textbf{Cls}\ \frac{P \xrightarrow{\overline{x}(y)\cdot(\nu y)} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} (\nu y)(P'|Q')}$$

$$\textbf{ClsSeq1}\ \frac{P \xrightarrow{\overline{x}(y)\cdot(\nu y)\cdot\sigma} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\sigma} (\nu y)(P'|Q')} \qquad \textbf{ClsSeq2}\ \frac{P \xrightarrow{\overline{x}(y)\cdot\gamma} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\gamma} P'|Q'}$$

γ does not start with a ν

$$opn(\overline{x}y, y) = \overline{x}(y)\cdot(\nu y) \qquad opn(\overline{x}y\cdot\sigma, y) = \begin{cases} \overline{x}(y)\cdot opn(\sigma, y) & if\ y \in obj(\sigma) \\ \overline{x}(y)\cdot(\nu y)\cdot opn(\sigma, y) & if\ y \notin obj(\sigma) \end{cases}$$

$$opn(\overline{x}z, y) = \overline{x}z \qquad opn(\overline{x}z\cdot\sigma, y) = \overline{x}z\cdot opn(\sigma, y)$$

$$opn((\nu z), y) = (\nu z) \qquad opn((\nu z)\cdot\sigma, y) = (\nu z)\cdot opn(\sigma, y)$$

$$sbj(\tau) = \emptyset \quad sbj(\overline{x}y) = \{x\} \quad sbj(x(y)) = \{x\} \quad sbj((\nu y)) = \emptyset \quad sbj(\alpha\cdot\sigma) = sbj(\alpha) \cup sbj(\sigma)$$

$$obj(\tau) = \emptyset \quad obj(\overline{x}y) = \{y\} \quad obj(x(y)) = \{y\} \quad obj((\nu y)) = \emptyset \quad obj(\alpha\cdot\sigma) = obj(\alpha) \cup obj(\sigma)$$

Table 3.4: Multi π late semantic: scope extrusion rules

we can only infer

$$(\nu x)(\underline{\overline{y}x}.\underline{\overline{a}b}.\underline{\overline{y}x}.\overline{a}b.P)|y(z).Q|a(b).R|y(z).S|a(b).T \xrightarrow{\tau} (\nu x)((\nu x)(P|Q\{x/z\})|R|S\{x/z\})|T$$

This transition is not what we want because now the scope of the inner νx hides in P the scope of the outer νx, so P and S cannot use x to communicate. But with the rules of table 3.4 the following transition can be inferred:

$$(\nu x)(\underline{\overline{y}x}.\underline{\overline{a}b}.\underline{\overline{y}x}.\overline{a}b.P)|y(z).Q|a(b).R|y(z).S|a(b).T \xrightarrow{\tau} (\nu x)(P|Q\{x/z\}|R|S\{x/z\})|T$$

Example Scope intrusion without strong prefixing.

$$\overline{y}x.P|(\nu x)(y(z).Q) \xrightarrow{\tau} P|(\nu w)(Q\{w/x\}\{x/z\})$$

Example Scope extrusion without strong prefixing.

$$\textbf{Cls}\ \dfrac{\textbf{Opn}\ \dfrac{\textbf{Out}\ \dfrac{}{\overline{y}x.P \xrightarrow{\overline{y}x} P}}{(\nu x)(\overline{y}x.P) \xrightarrow{\overline{y}(x)(\nu x)} P} \qquad \textbf{EInp}\ \dfrac{}{y(z).Q \xrightarrow{yx} Q\{x/z\}}}{(\nu x)(\overline{y}x.P)|y(z).Q \xrightarrow{\tau} (\nu x)(P|Q\{x/z\})}$$

Example Scope extrusion with strong prefixing(2). $x \in fn(y(z).Q|a(b).R|y(z).S)$ and $x^{'} \notin fn(y(z).Q|a(b).R|y(z).S)$:

$$(\nu x)(\underline{\overline{y}x}.\underline{\overline{a}b}.\underline{\overline{y}x}.\overline{a}b.P)|y(z).Q|a(b).R|y(z).S|a(b).T \xrightarrow{\tau} (\nu x^{'})(P\{x^{'}/x\}|Q\{x^{'}/z\}|R|S\{x^{'}/z\})|T$$

Example Scope intrusion with strong prefixing.

$$\underline{\overline{y}x}.\overline{a}b.P|(\nu x)(y(z).Q)|(\nu b)(a(c).R) \xrightarrow{\tau} P|(\nu w)(Q\{w/x\}\{x/z\})|(\nu d)(R\{d/b\}\{b/c\})$$

In the following section we will try to prove that strong early bisimulation is preserved by some operators. Let $\twoheadrightarrow$ be the smallest relation induced by the rules in table 3.3 excluding the rule *Alp*, and by the rules in table 3.4. We would like to use the fact that: $P \xrightarrow{\gamma} Q$ imply $P \overset{\gamma}{\twoheadrightarrow} S$ and $S \equiv_\alpha Q$. But this does not hold because for example:

$$(\nu x)z(a).0 \equiv_\alpha (\nu y)z(a).0 \xrightarrow{zx} (\nu y)0 \quad (\nu x)z(a).0 \overset{zx}{\not\longrightarrow}$$

But we do have to following:

Lemma 3.2.7. If $P \xrightarrow{\gamma} Q$ then $P \equiv_\alpha R \overset{\gamma}{\twoheadrightarrow} S \equiv_\alpha Q$.

Proof. Similar to the proof of lemma 2.5.2 □

Lemma 3.2.8. Let $\twoheadrightarrow$ be the semantic in table 3.3 and 3.4 but without rule *Alp*. $P \overset{\gamma}{\twoheadrightarrow} Q$ and $b \notin n(P)$ imply $P\{b/a\} \xrightarrow{\gamma\{b/a\}} S$ and $S \equiv_\alpha Q\{b/a\}$.

Proof. By induction on the derivation of $P \overset{\gamma}{\twoheadrightarrow} Q$ and by cases on the last rule used in the derivation:

Out : $\overline{x}y.P \overset{\overline{x}y}{\twoheadrightarrow} P$. For definition $(\overline{x}y.P)\{b/a\} = \overline{x}\{b/a\}y\{b/a\}.(P\{b/a\})$ and for rule *Out*:

$$\overline{x}\{b/a\}y\{b/a\}.(P\{b/a\}) \overset{(\overline{x}y)\{b/a\}}{\twoheadrightarrow} P\{b/a\}$$

Tau similar.

SOut suppose for induction hypothesis that $P\{b/a\} \overset{\gamma\{b/a\}}{\twoheadrightarrow} Q\{b/a\}$ and that γ is a non empty sequence of outputs. Then for rule *SOut*: $\underline{\overline{x}y}\{b/a\}.P\{b/a\} \overset{\underline{\overline{x}y}\{b/a\}\cdot\gamma\{b/a\}}{\twoheadrightarrow} Q\{b/a\}$ which for definition of substitution imply

$$(\underline{\overline{x}y}.P)\{b/a\} \xrightarrow{(\underline{\overline{x}y}\cdot\gamma)\{b/a\}} Q\{b/a\}$$

EInp(1) let $y \neq b$, $y \neq a$ and $z \neq a$: $x(y).P \overset{xz}{\twoheadrightarrow} P\{z/y\}$. For definition $(x(y).P)\{b/a\} = x\{b/a\}(y).(P\{b/a\})$ and for rule *EInp*:

$$x\{b/a\}(y).P\{b/a\} \xrightarrow{x\{b/a\}z} P\{b/a\}\{z/y\}$$

and $P\{b/a\}\{z/y\} \equiv_\alpha P\{z/y\}\{b/a\}$

EInp(2) let $y \neq b$, $y \neq a$: $x(y).P \overset{xa}{\twoheadrightarrow} P\{a/y\}$. For definition $(x(y).P)\{b/a\} = x\{b/a\}(y).(P\{b/a\})$ and for rule *EInp*:

$$x\{b/a\}(y).P\{b/a\} \xrightarrow{x\{b/a\}a} P\{b/a\}\{a/y\}$$

and $P\{b/a\}\{a/y\} \equiv_\alpha P\{a/y\}\{b/a\}$

EInp(3) let $a \neq z$: $x(a).P \overset{xz}{\twoheadrightarrow} P\{z/a\}$. For definition $(x(a).P)\{b/a\} = x\{b/a\}(a).P$ and for rule *EInp*:

$$x\{b/a\}(a).P \xrightarrow{x\{b/a\}z} P\{z/a\}$$

and $P\{z/a\}\{b/a\} \equiv_\alpha P\{z/a\}$

EInp(4) $x(a).P \overset{xa}{\twoheadrightarrow} P$. For definition $(x(a).P)\{b/a\} = x\{b/a\}(a).P$ and for rule *EInp*:

$$x\{b/a\}(a).P \xrightarrow{x\{b/a\}b} P\{b/a\}.$$

and $P\{b/a\} \equiv_\alpha P\{b/a\}$

ParL :

$$\textbf{ParL}\ \frac{P \xrightarrow{\gamma} P' \qquad bn(\gamma) \cap fn(P) = \emptyset}{P + Q \xrightarrow{\gamma} P'}$$

for induction $P\{b/a\} \xrightarrow{\gamma\{b/a\}} S$ and $S \equiv_\alpha P'\{b/a\}$. For rule *Sum* and definition of substitution: $(P + Q)\{b/a\} \xrightarrow{\gamma\{b/a\}} P'\{b/a\}$ and the conclusion holds.

ParR similar.

SumL similar.

SumR similar.

ECom :

$$\textbf{ECom}\ \frac{P \xrightarrow{\overline{x}y} P' \qquad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

for induction on the first premise $P\{b/a\} \xrightarrow{(\overline{x}y)\{b/a\}} S_1$ and $S_1 \equiv_\alpha P'\{b/a\}$. For induction on the second premise $Q\{b/a\} \xrightarrow{(xy)\{b/a\}} S_2$ and $S_2 \equiv_\alpha Q'\{b/a\}$. For rule *ECom* and definition of substitution: $(P|Q)\{b/a\} \xrightarrow{\tau} S_1|S_2$ and $S_1|S_2 \equiv (P'|Q')\{b/a\}$.

EComSeq similar.

$Res(1)$ $x \neq a$:

$$\textbf{Res}\ \frac{P \overset{\gamma}{\twoheadrightarrow} P^{'} \qquad x \notin n(\gamma)}{(\nu x)P \overset{\gamma}{\twoheadrightarrow} (\nu x)P^{'}}$$

for induction $P\{b/a\} \xrightarrow{\gamma\{b/a\}} S$ and $S \equiv_\alpha P^{'}\{b/a\}$. For rule Res and definition of substitution: $(\nu x)P\{b/a\} \xrightarrow{\gamma\{b/a\}} (\nu x)S$ and $(\nu x)S \equiv_\alpha (\nu x)P^{'}\{b/a\}$.

$Res(2)$:

$$\textbf{Res}\ \frac{P \overset{\gamma}{\twoheadrightarrow} P^{'} \qquad a \notin n(\gamma)}{(\nu a)P \overset{\gamma}{\twoheadrightarrow} (\nu a)P^{'}}$$

the conclusion holds because $((\nu a)P)\{b/a\} = (\nu a)P$ and $\gamma\{b/a\} = \gamma$.

Ide :

$$\textbf{Ide}\ \frac{A(\tilde{x}) \overset{def}{=} P \qquad P\{\tilde{w}/\tilde{x}\} \overset{\gamma}{\twoheadrightarrow} P^{'}}{A(\tilde{x}) \overset{\gamma}{\twoheadrightarrow} P^{'}}$$

for induction $P\{\tilde{w}/\tilde{x}\}\{b/a\} \xrightarrow{\gamma\{b/a\}} S$ and $S \equiv_\alpha P^{'}\{b/a\}$

$Opn(1)$ $y \neq a$:

$$\textbf{Opn}\ \frac{P \overset{\overline{y}x}{\twoheadrightarrow} P^{'} \qquad y \neq x}{(\nu y)P \overset{\overline{y}(x)}{\twoheadrightarrow} P^{'}}$$

for induction $P\{b/a\} \xrightarrow{\overline{y}x} S$ and $S \equiv_\alpha P^{'}\{b/a\}$

$Opn(2)$:

$$\textbf{Opn}\ \frac{P \overset{\overline{a}x}{\twoheadrightarrow} P^{'} \qquad a \neq x}{(\nu a)P \overset{\overline{a}(x)}{\twoheadrightarrow} P^{'}}$$

for induction $P\{b/a\} \xrightarrow{\overline{b}x} S$ and $S \equiv_\alpha P^{'}\{b/a\}$. So for rule Opn: $(\nu b)P\{b/a\} \xrightarrow{\overline{b}(x)} S$ and for rule Alp: $(\nu a)P \xrightarrow{\overline{b}(x)} S$

Cls :

$$\textbf{Cls}\ \frac{P \overset{\overline{x}(y)\cdot(\nu y)}{\twoheadrightarrow} P^{'} \qquad Q \overset{xy}{\twoheadrightarrow} Q^{'}}{P|Q \overset{\tau}{\twoheadrightarrow} (\nu y)(P^{'}|Q^{'})}$$

for induction on the first premise $P\{b/a\} \xrightarrow{\overline{x}\{b/a\}(y)\cdot(\nu y)} S$ and $S \equiv_\alpha P^{'}\{b/a\}$. For induction on the second premise $Q\{b/a\} \xrightarrow{\overline{x}\{b/a\}(y)\cdot(\nu y)} T$ and $T \equiv_\alpha Q^{'}\{b/a\}$. For rule Cls and definition of substitution: $(P|Q)\{b/a\} \xrightarrow{\tau} (\nu y)(S|T)$ and $(\nu y)S|T \equiv_\alpha (\nu y)(P^{'}\{b/a\}|Q^{'}\{b/a\})$.

$ClsSeq1$ similar.

$ClsSeq2$ similar.

$\square$

3.3 Strong bisimilarity and equivalence

3.3.1 Strong bisimilarity

In the following section, $\rightarrow$ is the transition relation defined in table 3.3.

Definition 3.3.1. A *strong early bisimulation* is a simmetric binary relation $\mathbf{S}$ on multi π processes such that for all P and Q: $P\mathbf{S}Q$, $P \xrightarrow{\gamma} P^{'}$ and $bn(\gamma)$ is fresh imply that

$$\exists Q^{'} : Q \xrightarrow{\gamma} Q^{'} \text{ and } P^{'}\mathbf{S}Q^{'}$$

The *strong early bisimilarity*, written $\sim_E$, is the union of all strong early bisimulation. Two processes P, Q are *strong early bisimilar*, written $P \sim_E Q$, if they are related by the strong early bisimilarity. The strong early bisimilarity is a strong early bisimulation.

Definition 3.3.2. A *strong early bisimulation up to* $\sim_E$ is a simmetric binary relation $\mathbf{S}$ on multi π processes such that for all P and Q: $P\mathbf{S}Q$, $P \xrightarrow{\gamma} P^{'}$ and $bn(\gamma)$ is fresh imply that

$$\exists P^{''}, Q^{'}, Q^{''} : Q \xrightarrow{\gamma} Q^{'} \text{ and } P^{'} \sim_E P^{''}\mathbf{S}Q^{''} \sim_E Q^{'}$$

Two processes P, Q are *strong early bisimilar up to* $\sim_E$, written $P \sim_E^{up} Q$, if they are related by a strong early bisimulation up to $\sim_E$.

Definition 3.3.3. A *strong early bisimulation up to restriction* is a simmetric binary relation $\mathbf{S}$ on multi π processes such that for all P and Q: $P\mathbf{S}Q$ imply

- for all $w \notin (fn(P) \cup fn(Q))$: $P\{w/z\} \sim_E Q\{w/z\}$
- $P \xrightarrow{\gamma} P^{'}$ and γ is not a τ imply there exists $Q^{'}$ such that $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\mathbf{S}Q^{'}$
- $P \xrightarrow{\tau} P^{'}$ then for some $Q^{'}$: $Q \xrightarrow{\tau} Q^{'}$ and either $P^{'}\mathbf{S}Q^{'}$ or for some $P^{''}, Q^{''}$ and w: $P^{'} \equiv (\nu w)P^{''}$, $Q^{'} \equiv (\nu w)Q^{''}$ and $P^{''}\mathbf{S}Q^{''}$

Two processes P, Q are *strong early bisimilar up to restriction*, written $P \sim_E^{\nu} Q$, if they are related by a strong early bisimulation up to restriction.

3.3.2 Properties of strong early bisimilarity

Proposition 3.3.1. $\sim_E$ is an equivalence relation.

Proof. :

Reflexivity The identity relation on processes is a strong early bisimulation.

Simmetry It is in the definition.

Transitivity The composition $\sim_E\sim_E$ is a strong early bisimulation.

□

Proposition 3.3.2. $P \sim_E^{up} Q$ imply $P \sim_E Q$.

Proof. Let $\mathbf{S}$ be a bisimulation up to $\sim_E$ such that $P\mathbf{S}Q$. It can be proved that $\sim_E \mathbf{S} \sim_E$ is a bisimulation: let $A \sim_E B\mathbf{S}C \sim_E D$

$$\begin{aligned}
&A \xrightarrow{\gamma} A^{'} \wedge A \sim_E B \wedge \text{definition 3.3.1} \Rightarrow \exists B^{'} : B \xrightarrow{\gamma} B^{'} \wedge A^{'} \sim_E B^{'}\\
&B\mathbf{S}C \wedge \text{definition 3.3.2} \Rightarrow \exists C^{'}C^{''}B^{''} : C \xrightarrow{\gamma} C^{'} \wedge B^{'} \sim_E B^{''}\mathbf{S}C^{''} \sim_E C^{'}\\
&C \xrightarrow{\gamma} C^{'} \wedge C \sim_E D \wedge \text{definition 3.3.1} \Rightarrow \exists D^{'} : D \xrightarrow{\gamma} D^{'} \wedge C^{'}\mathbf{S}D^{'}\\
&A^{'} \sim_E B^{'} \sim_E B^{''}\mathbf{S}C^{''} \sim_E C^{'} \sim_E D^{'} \wedge \text{transitivity of } \sim_E \Rightarrow A^{'} \sim_E B^{''}\mathbf{S}C^{''} \sim_E D^{'}
\end{aligned}$$

It is easy to see that the simmetric also holds. □

Proposition 3.3.3. If $\mathbf{S}$ is a strong early bisimulation up to restriction then $\mathbf{S} \subseteq \sim_E$.

Proof. Let $\mathbf{S}$ be a strong early bisimulation up to restriction then we define

$$\begin{cases} \mathbf{S}_0 = \mathbf{S} \\ \mathbf{S}_{n+1} = \{((\nu w)P, (\nu w)Q) : P\mathbf{S}_n Q, w \in \mathbf{N}\} \\ \mathbf{S}^* = \bigcup_{n<\omega} \mathbf{S}_n \end{cases}$$

Clearly $\mathbf{S} \subseteq \mathbf{S}^*$. We have to prove that $\mathbf{S}^*$ is a strong early bisimulation. The proof is an induction on n $\square$

Proposition 3.3.4. $\equiv_\alpha$ is a strong early bisimulation.

Proof. We prove by induction on the derivation of $P \equiv_\alpha Q$ and the derivation of $P \xrightarrow{\gamma} P'$ that $Q \xrightarrow{\gamma} Q'$ and $Q' \equiv_\alpha P'$. This imply that also the symmetric holds because α equivalence is symmetric. The last pair of rules used can be:

$(AlpOut, Out)$:

$$\textbf{AlpOut}\ \frac{P \equiv_\alpha Q}{\overline{x}y.P \equiv_\alpha \overline{x}y.Q} \qquad \textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{Out}\ \frac{}{\overline{x}y.Q \xrightarrow{\overline{x}y} Q}$$

$(AlpSOut, SOut)$:

$$\textbf{AlpSOut}\ \frac{P \equiv_\alpha Q}{\underline{\overline{x}y}.P \equiv_\alpha \underline{\overline{x}y}.Q} \qquad \textbf{SOut}\ \frac{P \xrightarrow{\gamma} P'}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\gamma} P'}$$

$Q \equiv_\alpha P \xrightarrow{\gamma} P'$ imply for inductive hypothesis that $Q \xrightarrow{\gamma} Q' \equiv P'$. So for rule $SOut$: $\underline{\overline{x}y}.Q \xrightarrow{\overline{x}y\cdot\gamma} Q'$

$(AlpTau, Tau)$:

$$\textbf{AlpTau}\ \frac{P \equiv_\alpha Q}{\tau.P \equiv_\alpha \tau.Q} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textbf{Tau}\ \frac{}{\tau.Q \xrightarrow{\tau} Q}$$

$(AlpInp, EInp)$:

$$\textbf{AlpInp}\ \frac{P \equiv_\alpha Q}{x(y).P \equiv_\alpha x(y).Q} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{EInp}\ \frac{}{x(y).Q \xrightarrow{xz} Q\{z/y\}}$$

$P \equiv_\alpha Q$ imply for lemma 2.3.6: $P\{z/y\} \equiv_\alpha Q\{z/y\}$

$(AlpInp1, EInp)$:

$$\textbf{AlpInp1}\ \frac{P \equiv_\alpha Q}{x(y).P \equiv_\alpha x(z).Q\{z/y\}} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xw} P\{w/y\}}$$

For rule $EInp$: $x(z).Q\{z/y\} \xrightarrow{xw} Q\{z/y\}\{w/z\}$. $P \equiv_\alpha Q$ imply for lemma 2.3.6: $P\{w/y\} \equiv_\alpha Q\{w/y\}$, for transitivity $P\{w/y\} \equiv_\alpha Q\{z/y\}\{w/z\}$

$(AlpInp2, EInp)$ similar.

$(AlpIde, Ide)$ in this case P and Q are both equal to some identifier.

$(AlpPar, Par)$:

$$\textbf{AlpPar}\ \frac{P_1 \equiv_\alpha Q_1 \qquad P_2 \equiv_\alpha Q_2}{P_1|P_2 \equiv_\alpha Q_1|Q_2} \qquad \textbf{Par}\ \frac{P_1 \xrightarrow{\gamma} P_1'}{P_1|P_2 \xrightarrow{\gamma} P_1'}$$

For induction $Q_1 \xrightarrow{\gamma} Q_1^{'}$ and $P_1^{'} \equiv_\alpha Q_1^{'}$. For rule Par: $Q_1|Q_2 \xrightarrow{\gamma} Q_1^{'}$.

$(AlpPar, ECom)$:

$$\textbf{AlpPar}\ \frac{P_1 \equiv_\alpha Q_1 \qquad P_2 \equiv_\alpha Q_2}{P_1|P_2 \equiv_\alpha Q_1|Q_2} \qquad \textbf{ECom}\ \frac{P_1 \xrightarrow{\overline{x}y} P_1^{'} \qquad P_2 \xrightarrow{xy} P_2^{'}}{P_1|P_2 \xrightarrow{\tau} P_1^{'}|P_2^{'}}$$

For induction on the first premise of $ECom$: $Q_1 \xrightarrow{\overline{x}y} Q_1^{'}$ and $P_1^{'} \equiv_\alpha Q_1^{'}$. For induction on the second premise: $Q_2 \xrightarrow{xy} Q_2^{'}$ and $P_2^{'} \equiv_\alpha Q_2^{'}$. For rule $ECom$: $Q_1|Q_2 \xrightarrow{\tau} Q_1^{'}|Q_2^{'}$ and for rule $AlpPar$: $P_1^{'}|P_2^{'} \equiv_\alpha Q_1^{'}|Q_2^{'}$.

$(AlpPar, EComSeq)$ similar.

$(AlpPar, Cls)$:

$$\textbf{AlpPar}\ \frac{P_1 \equiv_\alpha Q_1 \qquad P_2 \equiv_\alpha Q_2}{P_1|P_2 \equiv_\alpha Q_1|Q_2} \qquad \textbf{Cls}\ \frac{P_1 \xrightarrow{\overline{x}(y)\cdot(\nu x)} P_1^{'} \qquad P_2 \xrightarrow{xy} P_2^{'}}{P_1|P_2 \xrightarrow{\tau} (\nu x)(P_1^{'}|P_2^{'})}$$

For induction on the first premise of $ECom$: $Q_1 \xrightarrow{\overline{x}(y)\cdot(\nu x)} Q_1^{'}$ and $P_1^{'} \equiv_\alpha Q_1^{'}$. For induction on the second premise: $Q_2 \xrightarrow{xy} Q_2^{'}$ and $P_2^{'} \equiv_\alpha Q_2^{'}$. For rule Cls: $Q_1|Q_2 \xrightarrow{\tau} (\nu x)(Q_1^{'}|Q_2^{'})$ and for rules $AlpPar$ and $AlpRes$: $(\nu x)(P_1^{'}|P_2^{'}) \equiv_\alpha (\nu x)(Q_1^{'}|Q_2^{'})$.

$(AlpPar, ClsSeq1)$ similar.

$(AlpPar, ClsSeq2)$ similar.

$(AlpRes, Res)$:

$$\textbf{AlpRes}\ \frac{P \equiv_\alpha Q}{(\nu y)P \equiv_\alpha (\nu y)Q} \qquad \textbf{Res}\ \frac{P \xrightarrow{\gamma} P^{'} \qquad y \notin n(\gamma)}{(\nu y)P \xrightarrow{\gamma} (\nu y)P^{'}}$$

$P \equiv_\alpha Q$ and $P \xrightarrow{\gamma} P^{'}$ imply for inductive hypothesis that $Q \xrightarrow{\gamma} Q^{'}$ and $Q^{'} \equiv_\alpha P^{'}$. For rule Res and $AlpRes$: $(\nu y)Q \xrightarrow{\gamma} (\nu y)Q^{'}$ and $(\nu y)Q^{'} \equiv_\alpha (\nu y)P^{'}$.

$(AlpRes, Opn)$ similar.

$(AlpRes1, Res)$:

$$\textbf{AlpRes1}\ \frac{P \equiv_\alpha Q}{(\nu y) \equiv_\alpha (\nu z)Q\{z/y\}} \qquad \textbf{Res}\ \frac{P \xrightarrow{\gamma} P^{'} \qquad y \notin n(\gamma)}{(\nu y)P \xrightarrow{\gamma} (\nu y)P^{'}}$$

$P \equiv_\alpha Q$ and $P \xrightarrow{\gamma} P^{'}$ imply for inductive hypothesis that $Q \xrightarrow{\gamma} Q^{'}$ and $Q^{'} \equiv_\alpha P^{'}$. So:

$$\textbf{Alp}\ \frac{\textbf{AlpRes2}\ \dfrac{Q \equiv_\alpha Q}{(\nu z)Q\{z/y\} \equiv_\alpha (\nu y)Q} \qquad \textbf{Res}\ \dfrac{Q \xrightarrow{\gamma} Q^{'} \qquad y \notin n(\gamma)}{(\nu y)Q \xrightarrow{\gamma} (\nu y)Q^{'}}}{(\nu z)Q\{z/y\} \xrightarrow{\gamma} (\nu y)Q^{'}}$$

and $Q^{'} \equiv_\alpha P^{'}$ imply for rule Res that $(\nu y)Q^{'} \equiv_\alpha (\nu y)P^{'}$.

$(AlpRes1, Opn)$ similar.

$(AlpRes2, Res)$ similar.

$(AlpRes2, Opn)$ similar.

$\square$

Lemma 3.3.5. $\sim_E$ is preserved by all operators except input prefixing.

Proof. The proof goes by cases on operators:

Output prefixing

The relation $\{(\overline{x}y.P, \overline{x}y.Q) : P \sim_E Q\} \cup \sim_E$ is a strong early bisimulation. We can apply the following rules to $\overline{x}y$:

Out $\overline{x}y.P \xrightarrow{\overline{x}y} P \sim_E Q \xleftarrow{\overline{x}y} \overline{x}y.Q$

Alp :

$$\textbf{Alp}\ \frac{\dfrac{P \equiv_\alpha R}{\overline{x}y.P \equiv_\alpha \overline{x}y.R} \qquad \overline{x}y.R \xrightarrow{\overline{x}y} R}{\overline{x}y.P \xrightarrow{\overline{x}y} R}$$

$\overline{x}y.P \xrightarrow{\overline{x}y} R$, $\overline{x}y.Q \xrightarrow{\overline{x}y} Q$ and $R \equiv_\alpha P \sim_E Q$ imply $\overline{x}y.P$ and $\overline{x}y.Q$ are early bisimilar up to α equivalence. For proposition 3.3.2: $\overline{x}y.P$ and $\overline{x}y.Q$ are early bisimilar.

Strong output prefixing

The relation $\{(\underline{\overline{x}y}.P, \underline{\overline{x}y}.Q) : P \sim_E Q\} \cup \sim_E$ is a strong early bisimulation: there are three cases to consider:

- If there exists a transition $P \xrightarrow{\gamma} P'$ where γ is a non empty sequence of outputs then we can apply the rule *SOut*:

$$\frac{P \xrightarrow{\gamma} P'}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\gamma} P'}$$

  $P \xrightarrow{\gamma} P'$ and $P \sim_E Q$ imply $Q \xrightarrow{\gamma} Q'$ and $P' \sim_E Q'$. For rule *SOut*: $\underline{\overline{x}y}.Q \xrightarrow{\overline{x}y\cdot\gamma} Q'$ so the conclusion holds.

- There exists a process R α equivalent to P such that $R \xrightarrow{\gamma} R'$ where γ is a non empty sequence of outputs. We can apply the following rules:

$$\textbf{Alp}\ \frac{\dfrac{P \equiv_\alpha R}{\underline{\overline{x}y}.P \equiv_\alpha \underline{\overline{x}y}.R} \qquad \textbf{SOut}\ \dfrac{R \xrightarrow{\gamma} R'}{\underline{\overline{x}y}.R \xrightarrow{\overline{x}y\cdot\gamma} R'}}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\gamma} R'}$$

  For rule *Alp*: $P \xrightarrow{\gamma} R'$ and so we are back to the previous case.

- Otherwise there is no transition starting from $\underline{\overline{x}y}.P$ or from $\underline{\overline{x}y}.Q$ so these processes are strongly bisimilar.

Tau prefixing

The relation $\{(\tau.P, \tau.Q) : P \sim_E Q\} \cup \sim_E$ is a strong early bisimulation. We have to consider in turn each rule that can be applied to $\tau.P$:

Tau $\tau.P \xrightarrow{\tau} P \sim_E Q \xleftarrow{\tau} \tau.Q$

Alp :

$$\textbf{Alp}\ \frac{\dfrac{P \equiv_\alpha R}{\tau.P \equiv_\alpha \tau.R} \qquad \tau.R \xrightarrow{\tau} R}{\tau.P \xrightarrow{\tau} R}$$

$\tau.P \xrightarrow{\tau} R \equiv_\alpha P \sim_E Q$ and $\tau.Q \xrightarrow{\tau} Q$ imply $\tau.P$ and $\tau.Q$ are early bisimilar up to α equivalence. For proposition 3.3.2: $\tau.P$ and $\tau.Q$ are early bisimilar.

Sum

The relation $\{(P+R, Q+R) : P \sim_E Q\} \cup \sim_E$ is a strong early bisimulation. The rules that can be applied to $P+Q$ are:

Sum $P+R \xrightarrow{\gamma} P' \sim_E Q' \xleftarrow{\gamma} Q+R$

Alp :

$$\textbf{Alp}\ \dfrac{\dfrac{P \equiv_\alpha P_1 \qquad R \equiv_\alpha R_1}{P+R \equiv_\alpha P_1+R_1} \qquad \textbf{Sum}\ \dfrac{P_1 \xrightarrow{\gamma} P_1'}{P_1+R_1 \xrightarrow{\gamma} P_1'}}{P+R \xrightarrow{\gamma} P_1'}$$

$P \equiv_\alpha P_1$ and $P_1 \xrightarrow{\gamma} P_1'$ imply for rule *Alp*: $P \xrightarrow{\gamma} P_1'$ which in turn imply $Q \xrightarrow{\gamma} Q_1'$ and $P_1' \sim_E Q_1'$ since $P \sim_E Q$. Now an application of the rule *Sum* yields $Q+R \xrightarrow{\gamma} Q_1'$.

Restriction

We prove that the relation $Res(\sim_E) = \{((\nu x)P, (\nu x)Q) : P \sim_E Q\} \cup \sim_E$ is a strong early bisimulation up to α equivalence. This imply that $Res(\sim_E)$ is a strong early bisimulation. For lemma 3.2.7 and for reflexivity of α equivalence we can assume that every transition that starts from $(\nu x)P$ is in the form $(\nu x)P \equiv_\alpha R$ and $R \overset{\gamma}{\twoheadrightarrow} R'$. Where $\twoheadrightarrow$ is the semantic in table 3.3 and 3.4 but without rule *Alp*. So we can proceed by induction on both the derivation of $(\nu x)P \equiv_\alpha R$ and of $R \overset{\gamma}{\twoheadrightarrow} R'$. There are some cases depending on the last pair of rule used in the derivation of $(\nu x)P \equiv_\alpha R$ and of $R \overset{\gamma}{\twoheadrightarrow} R'$:

$(AlpRes, Res)$:

$$\textbf{Alp}\ \dfrac{\textbf{AlpRes}\ \dfrac{P \equiv_\alpha R}{(\nu x)P \equiv_\alpha (\nu x)R} \qquad \textbf{Res}\ \dfrac{R \overset{\gamma}{\twoheadrightarrow} R' \qquad x \notin n(\gamma)}{(\nu x)R \overset{\gamma}{\twoheadrightarrow} (\nu x)R'}}{(\nu x)P \overset{\gamma}{\twoheadrightarrow} (\nu x)R'}$$

$P \equiv_\alpha R \overset{\gamma}{\twoheadrightarrow} R'$ imply $P \xrightarrow{\gamma} R'$ which together with $P \sim_E Q$ imply $Q \xrightarrow{\gamma} Q'$ and $Q' \sim_E R'$. For rule *Res*: $(\nu x)Q \xrightarrow{\gamma} (\nu x)Q'$. Putting it all together:

$$(\nu x)P \xrightarrow{\gamma} (\nu x)P' \equiv_\alpha (\nu x)R'\, Res(\sim_E)(\nu x)Q' \xleftarrow{\gamma} (\nu x)Q$$

$(AlpRes, Res1)$: let $(\nu x)P \xrightarrow{\gamma} P'$, having in mind lemma 3.2.7

$$\textbf{Alp}\ \dfrac{\textbf{AlpRes1}\ \dfrac{P \equiv_\alpha R \qquad y \notin n(R)}{(\nu x)P \equiv_\alpha (\nu y)R\{y/x\}} \qquad \textbf{Res}\ \dfrac{R\{y/x\} \overset{\gamma}{\twoheadrightarrow} R' \qquad y \notin n(\gamma)}{(\nu y)R\{y/x\} \overset{\gamma}{\twoheadrightarrow} (\nu y)R'}}{(\nu x)P \overset{\gamma}{\twoheadrightarrow} (\nu y)R'}$$

$R\{y/x\} \overset{\gamma}{\twoheadrightarrow} R'$	lemma 3.2.8
imply $R \overset{\gamma\{x/y\}}{\twoheadrightarrow} S$ and $S \equiv_\alpha R'\{x/y\}$	$P \equiv_\alpha R$ and rule *Alp*
imply $P \xrightarrow{\gamma\{x/y\}} S$.	$P \sim_E Q$
imply $Q \xrightarrow{\gamma\{x/y\}} Q'$ and $Q' \sim_E S$.	

$Q \xrightarrow{\gamma\{x/y\}} Q'$ lemma 3.2.8
imply $Q\{y/x\} \xrightarrow{\gamma} Q'\{y/x\}$. rule *Res*
imply $(\nu y)Q\{y/x\} \xrightarrow{\gamma} (\nu y)Q'\{y/x\}$ rule *Alp* and $(\nu x)Q \equiv_\alpha (\nu y)Q\{y/x\}$
imply $(\nu x)Q \xrightarrow{\gamma} (\nu y)Q'\{y/x\}$

$Q' \sim_E R'\{x/y\}$
imply $(\nu x)Q'\, Res(\sim_E)(\nu x)R'\{x/y\}$

Putting it all together:

$$(\nu x)P \xrightarrow{\gamma} (\nu y)R' \equiv_\alpha (\nu x)R'\{x/y\} \equiv_\alpha (\nu x)S\ Res(\sim_E)\ (\nu x)Q' \equiv_\alpha (\nu y)Q'\{y/x\} \xleftarrow{\gamma} (\nu x)Q$$

$(AlpRes, Res2)$ similar.

$(AlpRes, Opn)$ similar.

$(AlpRes1, Opn)$ similar.

$(AlpRes2, Opn)$ similar.

Parallel composition

We prove that the relation $Par(\sim_E) = \{(P|R, Q|R) : P \sim_E Q\} \cup \sim_E$ is a strong early bisimulation up to α equivalence. This imply that $Par(\sim_E)$ is a strong early bisimulation. For lemma 3.2.7 and for reflexivity of α equivalence we can assume that every transition that starts from $P|R$ is in the form $P|R \equiv_\alpha S$ and $S \overset{\gamma}{\twoheadrightarrow} S'$. Where $\twoheadrightarrow$ is the semantic in table 3.3 and 3.4 but without rule *Alp*. We can proceed by induction on both the derivation of $P|R \equiv_\alpha Q|R$ and of $S \overset{\gamma}{\twoheadrightarrow} S'$. The last rule used in the derivation of $P|R \equiv_\alpha Q|R$ can only be *AlpPar*, but there are some different cases on the last rule used to prove that $S \overset{\gamma}{\twoheadrightarrow} S'$:

ECom :

$$\textbf{Alp}\ \frac{\textbf{Alp}\ \dfrac{P \equiv_\alpha P_2 \qquad R \equiv_\alpha R_2}{P|R \equiv_\alpha P_2|R_2} \qquad \textbf{ECom}\ \dfrac{P_2 \overset{\overline{x}y}{\twoheadrightarrow} P_2' \qquad R_2 \overset{xy}{\twoheadrightarrow} R_2'}{P_2|R_2 \overset{\tau}{\twoheadrightarrow} P_2'|R_2'}}{P|R \xrightarrow{\tau} P_2'|R_2'}$$

$P \equiv_\alpha P_2$ and $P_2 \xrightarrow{\overline{x}y} P_2'$ for rule *Alp* imply $P \xrightarrow{\overline{x}y} P_2'$. $P \sim_E Q$ imply that there exists a process Q' such that $Q \xrightarrow{\overline{x}y} Q'$ and $P_2' \sim_E Q'$. $R \equiv_\alpha R_2$ and $R_2 \overset{xy}{\twoheadrightarrow} R_2'$ imply for rule *Alp* that $R \overset{xy}{\twoheadrightarrow} R_2'$. So for rule *ECom*: $Q|R \xrightarrow{\tau} Q'|R_2'$ and $P_2'|R_2' \sim_E Q'|R_2'$

Cls :

$$\frac{P \xrightarrow{\overline{x}(y)\cdot(\nu y)} P' \qquad R \xrightarrow{xy} R'}{P|R \xrightarrow{\tau} (\nu y)(P'|R')}$$

$P \xrightarrow{\overline{x}(y)\cdot(\nu y)} P'$ and $P \sim_E Q$ imply that there exists a process Q' such that $Q \xrightarrow{\overline{x}y\cdot(\nu y)} Q'$ and $P' \sim_E Q'$. So for rule *Cls*: $Q|R \xrightarrow{\tau} (\nu y)(Q'|R')$ and $(\nu y)(P'|R') \sim_E (\nu y)(Q'|R')$

ClsSeq1, ClsSeq2, ParL, ParR similar.

□

Example $\sim_E$ is not in general preserved by input prefixing because:

$$a(x).0|\bar{b}y.0 \sim_E a(x).\bar{b}y.0 + \bar{b}y.a(x).0$$

but

$$c(a).(a(x).0|\bar{b}y.0) \dot{\not\sim}_E c(a).(a(x).\bar{b}y.0 + \bar{b}y.a(x).0)$$

because

$$\begin{array}{l} c(a).(a(x).0|\bar{b}y.0) \xrightarrow{cb} b(x).0|\bar{b}y.0 \xrightarrow{\tau} 0|0 \\ c(a).(a(x).\bar{b}y.0 + \bar{b}y.a(x).0) \xrightarrow{cb} b(x).\bar{b}y.0 + \bar{b}y.b(x).0 \stackrel{\tau}{\not\rightarrow} \end{array}$$

3.3.3 Strong D equivalence

Definition 3.3.4. A *distinction* is a finite symmetric and irreflexive binary relation on names. A substitution σ *respects* a pair (a, b) if

$$a\sigma \neq b\sigma$$

A substitution σ *respects* a distinction D if it respects every pair in the distinction:

$$\forall a, b.\ aDb \Rightarrow a\sigma \neq b\sigma$$

We write $D \cdot \sigma$ for the composition of the two relation.

Example The empty relation $\emptyset$ is a distinction. Every substitution respects the empty distinction.

Definition 3.3.5. Let D be a distinction and A be a set of names

$$D - A \stackrel{def}{=} D - (A \times \mathbb{N} \cup \mathbb{N} \times A)$$

Definition 3.3.6. Let D be a distinction and σ be a substitution. The application of σ to D is defined as:

$$D\sigma \stackrel{def}{=} \{(a\sigma, b\sigma) : (a, b) \in D\}$$

Proposition 3.3.6. Let $D, D^{'}$ be distinctions and σ be a substitution. Then

$$D^{'} \subseteq D \text{ and } \sigma \text{ respects } D \text{ imply } \sigma \text{ respects } D^{'}$$

Lemma 3.3.7. Let σ be a substitution, D be a distinction and $c \notin n(D)$. If σ respects D then $\sigma\{c/x\}$ respects $D - \{x\}$.

Proof. : σ respects D and $D - \{x\} \subseteq D$ imply σ respects $D - \{x\}$. $(d_1, d_2) \in (D - \{x\})$ imply $d_1\sigma\{c/x\} = d_1\sigma$ and $d_2\sigma = d_2\sigma\{c/x\}$. σ respects $D - \{x\}$ and $(d_1, d_2) \in (D - \{x\})$ for definition 3.3.4 imply $d_1\sigma \neq d_2\sigma$. Putting it all together $\sigma\{c/x\}$ respects (d_1, d_2). □

According to [2] the following holds:

Lemma 3.3.8. Let σ be a substitution, D be a distinction and $y\sigma = y$. If σ respects $D - \{x\}$ then $\{y/x\}\sigma$ respects D.

Definition 3.3.7. P and Q are *strongly D equivalent*, written $P \sim^D Q$, if for all substitution σ respecting D: $P\sigma \sim_E Q\sigma$. In this definition we assume that the application of σ to P and Q does not change any bound name.

Lemma 3.3.9. For any distinction D $\sim^D$ is an equivalence relation

Proof. $\sim^D$ is an equivalence relation because $\sim_E$ is an equivalence relation.

Reflexivity Since $\sim_E$ is reflexive, for all substitution σ respecting D: $P\sigma \sim_E Q\sigma$ so $P \sim^D P$

Symmetry Let $P \sim^D Q$ then for all substitution σ respecting D: $P\sigma \sim_E Q\sigma$. Since $\sim_E$ is symmetric $Q\sigma \sim_E P\sigma$ so $Q \sim^D P$

Transitivity Let $P \sim^D Q$ and $Q \sim^D R$ then for all substitution σ respecting D: $P\sigma \sim_E Q\sigma$ and $Q\sigma \sim_E R\sigma$. Since $\sim_E$ is transitive $P\sigma \sim_E R\sigma$ so $P \sim^D R$.

$\square$

Lemma 3.3.10. If $P \sim^D Q$ and for all $v \in fn(P,Q)$ such that $(v,y) \in D$ it holds that $P\{v/y\} \sim^D Q\{v/y\}$ then $x(y).P \sim^D x(y).Q$

Proof. DA RIGUARDARE ! ! Let σ be a substitution that respects D. If $y\sigma^{-1} = \{y\}$ then

$$(x(y).P)\sigma = x\sigma(y).P\sigma \xrightarrow{x\sigma z} P\sigma\{z/y\} \quad (x(y).Q)\sigma = x\sigma(y).Q\sigma \xrightarrow{x\sigma z} Q\sigma\{z/y\}$$

If $y \notin (y\sigma^{-1})$ then $(x(y).P)\sigma = x\sigma(w).P\{w/y\}\sigma \xrightarrow{x\sigma z} P\{w/y\}\sigma\{z/w\}$ where $w \notin n(x(y).P)$. $\square$

Lemma 3.3.11. If $P \sim^D Q$ then

- $\tau.P \sim^D \tau.Q$
- $\overline{x}y.P \sim^D \overline{x}y.Q$
- $\underline{\overline{x}y}.P \sim^D \underline{\overline{x}y}.Q$
- $P + R \sim^D Q + R$
- $P|R \sim^D Q|R$
- $(\nu x)P \sim^D (\nu x)Q$

Proof. $\sim^D$ is preserved by every operator. Let $P \sim^D Q$ and let σ be a substitution respecting D so $P\sigma \sim_E Q\sigma$:

Output prefixing

$P \sim^D Q$	definition 3.3.7
$\Rightarrow \forall\sigma$ respecting D. $P\sigma \sim_E Q\sigma$	lemma 3.3.5
$\Rightarrow (\overline{x}y)\sigma.(P\sigma) \sim_E (\overline{x}y)\sigma.(Q\sigma)$	definition of substitution
$\Rightarrow (\overline{x}y.P)\sigma \sim_E (\overline{x}y.Q)\sigma$	definition 3.3.7
$\Rightarrow \overline{x}y.P \sim^D \overline{x}y.Q$	

Strong output prefixing similar.

Tau prefixing

$P \sim^D Q$	definition 3.3.7
$\Rightarrow \forall\sigma$ respecting D. $P\sigma \sim_E Q\sigma$	lemma 3.3.5
$\Rightarrow \tau.(P\sigma) \sim_E \tau.(Q\sigma)$	definition of substitution
$\Rightarrow (\tau.P)\sigma \sim_E (\tau.Q)\sigma$	definition 3.3.7
$\Rightarrow \tau.P \sim^D \tau.Q$	

Sum

$P \sim^D Q$	definition 3.3.7
$\Rightarrow \forall\sigma$ respecting D. $P\sigma \sim_E Q\sigma$	lemma 3.3.5
$\Rightarrow (P\sigma) + (R\sigma) \sim_E (Q\sigma) + (R\sigma)$	definition of substitution
$\Rightarrow (P + R)\sigma \sim_E (Q + R)\sigma$	definition 3.3.7
$\Rightarrow P + R \sim^D Q + R$	

Parallel composition

$P \sim^D Q$	definition 3.3.7
$\Rightarrow \forall \sigma$ respecting D. $P\sigma \sim_E Q\sigma$	lemma 3.3.5
$\Rightarrow (P\sigma)\vert(R\sigma) \sim_E (Q\sigma)\vert(R\sigma)$	definition of substitution
$\Rightarrow (P\vert R)\sigma \sim_E (Q\vert R)\sigma$	definition 3.3.7
$\Rightarrow P\vert R \sim^D Q\vert R$	

Restriction Note that in definition 3.3.7 we assume that the substitution does not change any bound name so $((\nu x)P)\sigma = (\nu x)(P\sigma)$:

$P \sim^D Q$	definition 3.3.7
$\Rightarrow \forall \sigma$ respecting D. $P\sigma \sim_E Q\sigma$	lemma 3.3.5
$\Rightarrow (\nu x)(P\sigma) \sim_E (\nu x)(Q\sigma)$	definition of substitution
$\Rightarrow ((\nu x)P)\sigma \sim_E ((\nu x)Q)\sigma$	definition 3.3.7
$\Rightarrow (\nu x)P \sim^D (\nu x)Q$	

$\square$

Theorem 3.3.12. $\sim^{\emptyset}$ is a congruence.

Proof. Lemma 3.3.9 and put $D = \emptyset$ in lemma 3.3.11 and in lemma 3.3.10 $\square$

3.3.4 Open bisimulation

The following is an extension of the definition of strong open bisimulation found in [4]:

Definition 3.3.8. A *strong open bisimulation* is a symmetric binary relation $\mathbf{R}$ on multi π processes such that for all substitution σ:

Chapter 4

Multi π calculus with strong input

4.1 Syntax

As we did with π calculus, we suppose that we have a countable set of names $\mathbf{N}$, ranged over by lower case letters $a, b, \cdots, z$. This names are used for communication channels and values. Furthermore we have a set of identifiers, ranged over by A. We represent the agents or processes by upper case letters $P, Q, \cdots$. A multi π process, in addiction to the same actions of a π process, can perform also a strong prefix input:

$$\pi ::= \overline{x}y \mid x(z) \mid \underline{x(y)} \mid \tau$$

The process are defined, just as original π calculus, by the following grammar:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P + Q \mid (\nu x)P \mid A(y_1, \cdots, y_n)$$

and they have the same intuitive meaning as for the π calculus. The strong prefix input allows a process to make an atomic sequence of actions, so that more than one process can synchronize on this sequence. For the moment we allow the strong prefix to be on input names only. Also one can use the strong prefix only as an action prefixing for processes that can make at least a further action.

Multi π calculus is a conservative extension of the π calculus in the sense that: any π calculus process p is also a multi π calculus process and the semantic of p according to the SOS rules of π calculus is the same as the semantic of p according to the SOS rules of multi π calculus. We have to extend the following definition to deal with the strong prefix:

$$B(\underline{x(y)}.Q, I) = \{y, \overline{y}\} \cup B(Q, I) \qquad F(\underline{x(y)}.Q, I) = \{x, \overline{x}\} \cup (F(Q, I) - \{y, \overline{y}\})$$

The scope of the object of a strong input is the process that follows the strong input. For example the scope of a name x in a process $\underline{y(x)}.x(b).P$ is $x(b).P$.

In this setting two process cannot synchronize on a sequence of actions with length greater than one so we cannot have transactional synchronization but we can have multi-party synchronization.

4.2 Operational semantic

4.2.1 Early operational semantic with structural congruence

The semantic of a multi π process is labeled transition system such that

- the nodes are multi π calculus process. The set of node is $\mathbf{P}_m$
- the actions are multi π calculus actions. The set of actions is $\mathbf{A}_m$, we use $\alpha, \alpha_1, \alpha_2, \cdots$ to range over the set of actions, we use $\sigma, \sigma_1, \sigma_2, \cdots$ to range over the set $\mathbf{A}_m^+ \cup \{\tau\}$.
- the transition relations is $\rightarrow \subseteq \mathbf{P}_m \times (\mathbf{A}_m^+ \cup \{\tau\}) \times \mathbf{P}_m$

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{SInpTau}\ \frac{P\{y/z\} \xrightarrow{\tau} P'}{\underline{x(z)}.P \xrightarrow{xy} P'} \qquad \textbf{SInp}\ \frac{P\{y/z\} \xrightarrow{ab} P'}{\underline{x(z)}.P \xrightarrow{xy\cdot ab} P'} \qquad \textbf{SInpSeq}\ \frac{P\{y/z\} \xrightarrow{\sigma} P' \qquad |\sigma| > 1}{\underline{x(z)}.P \xrightarrow{xy\cdot\sigma} P'}$$

$$\textbf{Sum}\ \frac{P \xrightarrow{\sigma} P'}{P + Q \xrightarrow{\sigma} P'} \qquad \textbf{Cong}\ \frac{P \equiv P' \qquad P' \xrightarrow{\alpha} Q}{P \xrightarrow{\alpha} Q} \qquad \textbf{Res}\ \frac{P \xrightarrow{\sigma} P' \qquad z \notin n(\sigma)}{(\nu z)P \xrightarrow{\sigma} (\nu z)P'}$$

$$\textbf{Par}\ \frac{P \xrightarrow{\sigma} P'}{P|Q \xrightarrow{\sigma} P'|Q} \qquad \textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P' \qquad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P'}$$

$$\textbf{ECom}\ \frac{P \xrightarrow{xy} P' \qquad Q \xrightarrow{\overline{x}y} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad \textbf{EComSeq}\ \frac{P \xrightarrow{xy\cdot\sigma} P' \qquad Q \xrightarrow{\overline{x}y} Q'}{P|Q \xrightarrow{\sigma} P'|Q'}$$

Table 4.1: Multi π early semantic with structural congruence

In this case, a label can be a sequence of prefixes, whether in the original π calculus a label can be only a prefix. We use the symbol $\cdot$ to denote the concatenation operator.

Definition 4.2.1. The *early transition relation with structural congruence* is the smallest relation induced by the rules in table 4.1 where *inpSeq* is a non empty sequence of input actions and σ is a sequence of any action.

Example Multi-party synchronization We show an example of a derivation of three processes that synchronize.

$$\textbf{EComSeq}\ \frac{\textbf{SInp}\ \dfrac{\textbf{EInp}\ \dfrac{}{(x(b).P)\{y/a\} \xrightarrow{xz} P\{y/a\}\{z/b\}}}{\underline{x(a)}.(x(b).P) \xrightarrow{xy\cdot xz} P\{y/a\}\{z/b\}} \qquad \textbf{Out}\ \dfrac{}{\overline{x}y.Q \xrightarrow{\overline{x}y} Q}}{\underline{x(a)}.x(b).P|\overline{x}y.Q \xrightarrow{xz} P\{y/a\}\{z/b\}|Q}$$

$$\textbf{EComSng}\ \frac{\underline{x(a)}.x(b).P|\overline{x}y.Q \xrightarrow{xz} P\{y/a\}\{z/b\}|Q \qquad \textbf{Out}\ \dfrac{}{\overline{x}z.R \xrightarrow{\overline{x}z} R}}{(\underline{x(a)}.x(b).P|\overline{x}y.Q)|\overline{x}z.R \xrightarrow{\tau} (P\{y/a\}\{z/b\}|Q)|R}$$

Lemma 4.2.1. If $P \xrightarrow{\sigma} Q$ then only one of the following cases hold:

- $|\sigma| = 1$
- $|\sigma| > 1$, the actions in σ are input.

4.2.2 Late operational semantic with structural congruence

Definition 4.2.2. The *late transition relation with structural congruence* is the smallest relation induced by the rules in table 4.2.

Example Multi-party synchronization We show an example of a derivation of three processes that synchronize with the late semantic. The three processes are $\underline{x(a)}.x(b).P$, $\overline{x}y.Q$ and $\overline{x}z.R$. We assume modulo α conversion that:

$$a \notin fn(x(b)) \cup fn(\underline{x(a)}.x(b).P)$$

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textbf{LInp}\ \frac{}{x(y).P \xrightarrow{x(y)} P} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P}$$

$$\textbf{SInp}\ \frac{P \xrightarrow{\gamma} P^{'}}{\underline{x(z)}.P \xrightarrow{x(z)\cdot\gamma} P^{'}} \quad \gamma \text{ is a non empty sequence of inputs}$$

$$\textbf{LComSeq}\ \frac{P \xrightarrow{x(y)\cdot\sigma} P^{'} \qquad Q \xrightarrow{\overline{x}z} Q^{'} \qquad bn(\sigma)\cap fn(Q)=\emptyset}{P|Q \xrightarrow{\sigma\{z/y\}} P^{'}\{z/y\}|Q^{'}} \qquad \textbf{LCom}\ \frac{P \xrightarrow{x(y)} P^{'} \qquad Q \xrightarrow{\overline{x}z} Q^{'}}{P|Q \xrightarrow{\tau} P^{'}\{z/y\}|Q^{'}}$$

$$\textbf{Sum}\ \frac{P \xrightarrow{\sigma} P^{'}}{P+Q \xrightarrow{\sigma} P^{'}} \qquad \textbf{Cong}\ \frac{P \equiv P^{'} \qquad P^{'} \xrightarrow{\sigma} Q}{P \xrightarrow{\sigma} Q} \qquad \textbf{Opn}\ \frac{P \xrightarrow{\overline{x}z} P^{'} \qquad z \neq x}{(\nu z)P \xrightarrow{\overline{x}(z)} P^{'}}$$

$$\textbf{Res}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\sigma} (\nu z)P^{'}} \qquad \textbf{Par}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad bn(\sigma)\cap fn(Q)=\emptyset}{P|Q \xrightarrow{\sigma} P^{'}|Q}$$

Table 4.2: Multi π late semantic with structural congruence

and

$$c \notin fn(\overline{x}y.Q)$$

$$\textbf{LComSeq}\ \frac{\textbf{SInp}\ \dfrac{\textbf{LInp}\ \dfrac{}{x(b).P \xrightarrow{x(b)} P}}{\underline{x(a)}.x(b).P \xrightarrow{x(a)\cdot x(b)} P} \qquad \textbf{Out}\ \dfrac{}{\overline{x}y.Q \xrightarrow{\overline{x}y} Q}}{\underline{x(a)}.x(b).P|\overline{x}y.Q \xrightarrow{x(b)} P\{y/a\}|Q}$$

$$\textbf{LCom}\ \frac{\underline{x(a)}.x(b).P|\overline{x}y.Q \xrightarrow{x(b)} P\{y/a\}|Q \qquad \textbf{Out}\ \dfrac{}{\overline{x}z.R \xrightarrow{\overline{x}z} R}}{(\underline{x(a)}.x(b).P|\overline{x}y.Q)|\overline{x}z.R \xrightarrow{\tau} (P\{y/a\}|Q)\{z/b\}|R = (P\{y/a\}\{z/b\}|Q)|R}$$

4.2.3 Low level semantic

This section contains the definition of an alternative semantic for multi π. First we define a low level version of the multi π calculus(here with strong prefixing on input only), we call this language low multi π. The low multi π is the multi π enriched with a marked or intermediate process $*P$:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P+Q \mid (\nu x)P \mid A \mid *P$$

$$\pi ::= \overline{x}y \mid x(y) \mid \underline{x(y)} \mid \tau$$

Definition 4.2.3. The low level transition relation is the smallest relation induced by the rules in table 4.3 in which P stands for a process without mark, L stands for a process with mark and S can stand for both.

Lemma 4.2.2. For all unmarked processes P, Q and marked processes L_1, L_2.

- if $P \xmapsto{\alpha} L_1$ or $L_1 \xmapsto{\alpha} L_2$ then α can only be an input or an ϵ
- if $L_1 \xmapsto{\alpha} P$ then α is an input or a τ

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xmapsto{\overline{x}y} P} \qquad \textbf{EInp}\ \frac{}{x(y).P \xmapsto{xz} P\{z/y\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xmapsto{\tau} P}$$

$$\textbf{StarInp}\ \frac{P \xmapsto{xy} S'}{*P \xmapsto{xy} S'} \qquad \textbf{SInpLow}\ \frac{}{\underline{x(z)}.P \xmapsto{xy} *P\{y/z\}} \qquad \textbf{StarEps}\ \frac{P \xmapsto{\epsilon} S'}{*P \xmapsto{\epsilon} S'}$$

$$\textbf{Com1}\ \frac{P \xmapsto{\overline{x}y} P' \qquad Q \xmapsto{xy} Q'}{P|Q \xmapsto{\tau} P'|Q'}$$

$$\textbf{Com2L}\ \frac{L_1 \xmapsto{xy} L_2 \qquad P \xmapsto{\overline{x}y} Q}{L_1|P \xmapsto{\epsilon} L_2|Q} \qquad \textbf{Com2R}\ \frac{P \xmapsto{\overline{x}y} Q \qquad L_1 \xmapsto{xy} L_2}{P|L_1 \xmapsto{\epsilon} Q|L_2}$$

$$\textbf{Com3L}\ \frac{P \xmapsto{xy} L \qquad Q \xmapsto{\overline{x}y} Q'}{P|Q \xmapsto{\epsilon} L|Q'} \qquad \textbf{Com3R}\ \frac{Q \xmapsto{\overline{x}y} Q' \qquad P \xmapsto{xy} L}{Q|P \xmapsto{\epsilon} Q'|L}$$

$$\textbf{Com4L}\ \frac{L \xmapsto{xy} P \qquad Q \xmapsto{\overline{x}y} Q'}{L|Q \xmapsto{\tau} P|Q'} \qquad \textbf{Com4R}\ \frac{Q \xmapsto{\overline{x}y} Q' \qquad L \xmapsto{xy} P}{L|Q \xmapsto{\tau} P|Q'}$$

$$\textbf{Res}\ \frac{S \xmapsto{\gamma} S' \qquad y \notin n(\gamma)}{(\nu y)S \xmapsto{\gamma} (\nu y)S'} \qquad \textbf{Opn}\ \frac{P \xmapsto{\overline{x}y} Q \qquad y \neq x}{(\nu y)P \xmapsto{\overline{x}(y)} Q} \qquad \textbf{Cong}\ \frac{P \equiv P' \qquad P' \xmapsto{\gamma} S}{P \xmapsto{\gamma} S}$$

$$\textbf{Par1L}\ \frac{S \xmapsto{\gamma} S'}{S|Q \xmapsto{\gamma} S'|Q} \qquad \textbf{Par1R}\ \frac{S \xmapsto{\gamma} S'}{Q|S \xmapsto{\gamma} Q|S'} \qquad \textbf{Sum}\ \frac{P \xmapsto{\gamma} S}{P+Q \xmapsto{\gamma} S}$$

Table 4.3: Low multi π early semantic with structural congruence

- if $P \xmapsto{\alpha} Q$ then α is not an ϵ

Definition 4.2.4. Let P, Q be unmarked processes and $L_1, \cdots, L_{k-1}$ marked processes. We define the derivation relation $\rightarrow_s$ in the following way:

$$\textbf{Low}\ \dfrac{P \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} Q \qquad k \geq 1}{P \xrightarrow{\gamma_1 \cdots \gamma_k}_s Q}$$

We need to be precise about the concatenation operator $\cdot$ since we have introduced the new label ϵ. Let a be an action such that $a \neq \tau$ and $a \neq \epsilon$ then the following rules hold:

$$\begin{array}{lll} \epsilon \cdot a = a \cdot \epsilon = a & \epsilon \cdot \epsilon = \epsilon & \tau \cdot \epsilon = \epsilon \cdot \tau = \tau \\ \tau \cdot a = a \cdot \tau = a & \tau \cdot \tau = \tau & \end{array}$$

Example Multi-party synchronization We show an example of a derivation of three processes that synchronize.

$$\textbf{Par1L}\ \dfrac{\textbf{Com3L}\ \dfrac{\textbf{SInpLow}\ \dfrac{}{\underline{x(a)}.x(b).P \xmapsto{xy} *(x(b).P\{y/a\})} \qquad \textbf{Out}\ \dfrac{}{\overline{x}y.Q \xmapsto{\overline{x}y} Q}}{\underline{x(a)}.x(b).P|\overline{x}y.Q \xmapsto{\epsilon} *(x(b).P\{y/a\})|Q}}{(\underline{x(a)}.x(b).P|\overline{x}y.Q)|\overline{x}z.R \xmapsto{\epsilon} (*(x(b).P\{y/a\})|Q)|\overline{x}z.R}$$

$$\textbf{Par1L}\ \dfrac{\textbf{Star}\ \dfrac{\textbf{EInp}\ \dfrac{}{x(b).P\{y/a\} \xmapsto{xz} P\{y/a\}\{z/b\}}}{*(x(b).P\{y/a\}) \xmapsto{xz} P\{y/a\}\{z/b\}}}{*(x(b).P\{y/a\})|Q \xmapsto{xz} P\{y/a\}\{z/b\}|Q}$$

$$\textbf{Com4L}\ \dfrac{*(x(b).P\{y/a\})|Q \xmapsto{xz} P\{y/a\}\{z/b\}|Q \qquad \textbf{Out}\ \dfrac{}{\overline{x}z.R \xmapsto{\overline{x}z} R}}{(\underline{x(a)}.x(b).P|\overline{x}y.Q)|\overline{x}z.R \xmapsto{\tau} (P\{y/a\}\{z/b\}|Q)|R}$$

Proposition 4.2.3. Let $\rightarrow$ be the relation defined in table 4.1. If $P \xrightarrow{\sigma} Q$ then there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

Proof. The proof is by induction on the depth of the derivation tree of $P \xrightarrow{\sigma} Q$ and by cases on the last rule used in the derivation:

$EInp, Out, Tau$ These rules are also in table 4.3 so we can derive $P \xmapsto{\sigma} Q$.

$SInpSeq$ the last part of the derivation tree looks like this:

$$\textbf{SInpSeq}\ \dfrac{P_1\{y/z\} \xrightarrow{\sigma} Q \qquad |\sigma| > 1}{\underline{x(z)}.P_1 \xrightarrow{xy \cdot \sigma} Q}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1\{y/z\} \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

then a proof of the conclusion follows from:

$$\textbf{SInpLow}\ \dfrac{}{\underline{x(z)}.P_1 \xmapsto{xy} *P_1\{y/z\}} \qquad \textbf{Star}\ \dfrac{P_1\{y/z\} \xmapsto{\gamma_1} L_1}{*P_1\{y/z\} \xmapsto{\gamma_1} L_1}$$

where $Star$ means $StarInp$ or $StarEps$, note that γ_1 is an input or an *epsilon* because of 4.2.1.

SInp this case is similar to the previous.

SInpTau this case is similar to the previous observing that $xy \cdot \tau = xy$.

Sum the last part of the derivation tree looks like this:

$$\textbf{Sum}\ \frac{P_1 \xrightarrow{\sigma} Q}{P_1 + P_2 \xrightarrow{\sigma} Q}$$

for the inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

A proof of the conclusion is:

$$\textbf{Sum}\ \frac{P_1 \xmapsto{\gamma_1} L_1}{P_1 + P_2 \xmapsto{\gamma_1} L_1}$$

Cong this case is similar to the previous.

ECom the last part of the derivation tree looks like this:

$$\textbf{ECom}\ \frac{P_1 \xrightarrow{xy} P_1^{'} \qquad Q_1 \xrightarrow{\overline{x}y} Q_1^{'}}{P_1|Q_1 \xrightarrow{\tau} P_1^{'}|Q_1^{'}}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} P_1^{'} \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = xy$$

and there exist $R_1, \cdots, R_h$ and $\delta_1, \cdots, \delta_{h+1}$ with $h \geq 0$ such that

$$Q_1 \xmapsto{\delta_1} R_1 \xmapsto{\delta_2} R_2 \cdots R_{h-1} \xmapsto{\delta_h} R_h \xmapsto{\delta_{h+1}} Q_1^{'} \quad \text{and} \quad \delta_1 \cdot \ldots \cdot \delta_{h+1} = \overline{x}y$$

For lemma 4.2.2 there cannot be an output action in a transition involving marked processes so h must be 0 and $Q_1 \xmapsto{\delta_1} Q_1^{'}$ with $\delta_1 = \overline{x}y$. We can have three different cases now:

$\gamma_1 = xy$ A proof of the conclusion is:

$$P_1|Q_1 \xmapsto{\epsilon} L_1|Q_1^{'} \xmapsto{\epsilon} L_2|Q_1^{'} \cdots \xmapsto{\epsilon} L_k|Q_1^{'} \xmapsto{\tau} P_1^{'}|Q_1^{'}$$

we derive the first transition with rule *Com3L*, whether for the other transition we use the rule *Par1L*.

$\gamma_i = xy$ A proof of the conclusion is:

$$P_1|Q_1 \xmapsto{\epsilon} L_1|Q_1 \cdots \xmapsto{\epsilon} L_{i-1}|Q_1 \xmapsto{\epsilon} L_i|Q_1^{'} \xmapsto{\epsilon} L_{i+1}|Q_1^{'} \cdots \xmapsto{\epsilon} L_k|Q_1^{'} \xmapsto{\tau} P_1^{'}|Q_1^{'}$$

we derive the transaction $L_{i-1}|Q_1 \xmapsto{\epsilon} L_i|Q_1^{'}$ with rule *Com2L*, whether for the other transactions we use the rule *Par1L*.

$\gamma_{k+1} = xy$ similar.

Res the last part of the derivation tree looks like this:

$$\textbf{Res}\ \frac{P_1 \xrightarrow{\sigma} Q_1 \qquad z \notin n(\sigma)}{(\nu z)P_1 \xrightarrow{\sigma} (\nu z)Q_1}$$

for the inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q_1 \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

We can apply the rule *Res* to each of the previous transitions because

$$z \notin n(\sigma) \text{ implies } z \notin n(\gamma_i) \text{ for each } i$$

and then get a proof of the conclusion:

$$(\nu z)P_1 \overset{\gamma_1}{\longmapsto} (\nu z)L_1 \overset{\gamma_2}{\longmapsto} (\nu z)L_2 \cdots (\nu z)L_{k-1} \overset{\gamma_k}{\longmapsto} (\nu z)L_k \overset{\gamma_{k+1}}{\longmapsto} (\nu z)Q_1$$

Par this case is similar to the previous.

EComSeq the last part of the derivation tree looks like this:

$$\textbf{EComSeq}\ \frac{P_1 \xrightarrow{xy\cdot\sigma} P_1' \qquad Q_1 \xrightarrow{\overline{x}y} Q_1'}{P_1|Q_1 \xrightarrow{\sigma} P_1'|Q_1'}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} P_1' \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = xy \cdot \sigma$$

For inductive hypothesis and lemma 4.2.2 $Q_1 \overset{\overline{x}y}{\longmapsto} Q_1'$. We can have two different cases now depending on where the first xy is:

$\gamma_1 = xy$ A proof of the conclusion is:

$$P_1|Q_1 \overset{\epsilon}{\longmapsto} L_1|Q_1' \overset{\gamma_2}{\longmapsto} L_2|Q_1' \cdots \overset{\gamma_k}{\longmapsto} L_k|Q_1' \overset{\gamma_{k+1}}{\longmapsto} P_1'|Q_1'$$

we derive the first transition with rule *Com3L*, whether for the other transactions we use the rule *Par1L*. Since $\gamma_1 \cdot \ldots \cdot \gamma_{k+1} = xy \cdot \sigma$ and $\gamma_1 = xy$ then $\epsilon \cdot \gamma_2 \cdot \ldots \cdot \gamma_{k+1} = \sigma$

$\gamma_i = xy$ A proof of the conclusion is:

$$P_1|Q_1 \overset{\epsilon}{\longmapsto} L_1|Q_1 \cdots \overset{\epsilon}{\longmapsto} L_{i-1}|Q_1 \overset{\epsilon}{\longmapsto} L_i|Q_1' \overset{\gamma_{i+1}}{\longmapsto} L_{i+1}|Q_1' \cdots \overset{\gamma_k}{\longmapsto} L_k|Q_1' \overset{\gamma_{k+1}}{\longmapsto} P_1'|Q_1'$$

we derive the transition $L_{i-1}|Q_1 \overset{\epsilon}{\longmapsto} L_i|Q_1'$ with rule *Com2L*, whether for the other transactions of the premises we use the rule *Par1L*.

$\gamma_{k+1} = xy$ cannot happen because σ is not empty.

□

Proposition 4.2.4. Let $\rightarrow$ be the relation defined in table 4.1. Let α be an action. If $P \overset{\alpha}{\longmapsto} Q$ then $P \xrightarrow{\alpha} Q$.

Proof. The proof is by induction the depth of the derivation of $P \overset{\alpha}{\longmapsto} Q$:

Out, *EInp*, *Tau* These rules are also in table 4.1 so we can derive $P \xrightarrow{\alpha} Q$.

*Com*1

$$\textbf{Com1}\ \frac{P_1 \overset{xy}{\longmapsto} Q_1 \qquad P_2 \overset{\overline{x}y}{\longmapsto} Q_2}{P_1|P_2 \overset{\tau}{\longmapsto} Q_1|Q_2}$$

for inductive hypothesis $P_1 \xrightarrow{xy} Q_1$ and $P_2 \xrightarrow{\overline{x}y} Q_2$ so for rule *Com* $P_1|P_2 \xrightarrow{\tau} Q_1|Q_2$

Sum

$$\textbf{Sum}\ \frac{P_1 \overset{\alpha}{\longmapsto} Q}{P_1 + P_2 \overset{\alpha}{\longmapsto} Q}$$

for inductive hypothesis $P_1 \xrightarrow{\alpha} Q$ and for rule *Sum* $P_1 + P_2 \xrightarrow{\alpha} Q$.

Res the first transition is:

$$\textbf{Res}\ \frac{P_1 \overset{\alpha}{\longmapsto} Q_1 \qquad z \notin n(\gamma_1)}{(\nu z)P_1 \overset{\alpha}{\longmapsto} (\nu z)Q_1}$$

for inductive hypothesis $P_1 \xrightarrow{\alpha} Q_1$ and for rule *Res* $(\nu z)P_1 \xrightarrow{\alpha} (\nu z)Q_1$.

others other cases are similar.

□

4.3 Normal form

In the following section the symbol $\rightarrow$ will refer to the late semantic with structural congruence of multi π calculus with strong input which is illustrated in table 4.2. Also we consider a structural congruence without the rules $P|0 \equiv 0$ and $P + 0 \equiv 0$. For the purpose of clarity the rule of structural congruence are repeated in this secition.

Definition 4.3.1. $\twoheadrightarrow$ is the smallest relation induced by the all the rules in table 4.2 except *Cong*.

Proposition 4.3.1. If $P \xrightarrow{\sigma} Q$ then there exists a process R such that: $R \overset{\sigma}{\twoheadrightarrow} Q$ and $P \equiv R$

Proof. We show that we can move the rule *Cong* down the inference tree of $P \xrightarrow{\sigma} Q$. So a derivation of $P \xrightarrow{\sigma} Q$ can translate into a derivation of $P \xrightarrow{\sigma} Q$ which uses the rule *Cong* only as its last rule.

SInp

$$\textbf{SInp}\ \frac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\gamma} Q}{P \xrightarrow{\gamma} Q}}{\underline{x(z)}.P \xrightarrow{x(z)\cdot\gamma} Q}$$

become

$$\textbf{Cong}\ \frac{\dfrac{P \equiv R}{\underline{x(z)}.P \equiv \underline{x(z)}.R} \qquad \textbf{SInp}\ \dfrac{R \xrightarrow{\gamma} Q}{\underline{x(z)}.R \xrightarrow{x(z)\cdot\gamma} Q}}{\underline{x(z)}.P \xrightarrow{x(z)\cdot\gamma} Q}$$

Sum

$$\textbf{Sum}\ \frac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\gamma} Q}{P \xrightarrow{\gamma} Q}}{P + S \xrightarrow{\gamma} Q}$$

become

$$\textbf{Cong}\ \frac{\dfrac{P \equiv R}{P + S \equiv R + S} \qquad \textbf{Sum}\ \dfrac{R \xrightarrow{\gamma} Q}{R + S \xrightarrow{\gamma} Q}}{P + S \xrightarrow{\gamma} Q}$$

Cong

$$\textbf{Cong}\ \frac{P \equiv R \qquad \textbf{Cong}\ \dfrac{R \equiv S \qquad S \xrightarrow{\gamma} Q}{R \xrightarrow{\gamma} Q}}{P \xrightarrow{\gamma} Q}$$

become

$$\textbf{Cong}\ \frac{\dfrac{P \equiv R \qquad R \equiv S}{P \equiv S} \qquad S \xrightarrow{\gamma} Q}{P \xrightarrow{\gamma} Q}$$

Par

$$\textbf{Par}\ \frac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\gamma} Q}{P \xrightarrow{\gamma} Q} \qquad bn(\gamma) \cap fn(S) = \emptyset}{P|S \xrightarrow{\gamma} Q}$$

become

$$\textbf{Cong}\ \frac{\dfrac{P \equiv R}{P|S \equiv R|S} \qquad \textbf{Par}\ \dfrac{R \xrightarrow{\gamma} Q \qquad bn(\gamma) \cap fn(S) = \emptyset}{R|S \xrightarrow{\gamma} Q}}{P|S \xrightarrow{\gamma} Q}$$

LComSeq

$$\textbf{LComSeq}\ \frac{\textbf{Cong}\ \dfrac{P_1 \equiv R_1 \qquad R_1 \xrightarrow{x(y)\cdot\sigma} Q_1}{P_1 \xrightarrow{x(y)\cdot\sigma} Q_1} \qquad \textbf{Cong}\ \dfrac{P_2 \equiv R_2 \qquad R_2 \xrightarrow{\overline{x}z} Q_2}{P_2 \xrightarrow{\overline{x}z} Q_2}}{P_1|P_2 \xrightarrow{\gamma\{z/y\}} Q_1\{z/y\}|Q_2}$$

become

$$\textbf{Cong}\ \frac{\dfrac{P_1 \equiv R_1 \qquad P_2 \equiv R_2}{P_1|P_2 \equiv R_1|R_2} \qquad \textbf{LComSeq}\ \dfrac{R_1 \xrightarrow{x(y)\cdot\sigma} Q_1 \qquad R_2 \xrightarrow{\overline{x}z} Q_2}{R_1|R_2 \xrightarrow{\sigma\{z/y\}} Q_1\{z/y\}|Q_2}}{P_1|P_2 \xrightarrow{\gamma\{z/y\}} Q_1\{z/y\}|Q_2}$$

LCom

$$\textbf{LCom}\ \frac{\textbf{Cong}\ \dfrac{P_1 \equiv R_1 \qquad R_1 \xrightarrow{x(y)} Q_1}{P_1 \xrightarrow{x(y)} Q_1} \qquad \textbf{Cong}\ \dfrac{P_2 \equiv R_2 \qquad R_2 \xrightarrow{\overline{x}z} Q_2}{P_2 \xrightarrow{\overline{x}z} Q_2}}{P_1|P_2 \xrightarrow{\tau} Q_1\{z/y\}|Q_2}$$

become

$$\textbf{Cong}\ \frac{\dfrac{P_1 \equiv R_1 \qquad P_2 \equiv R_2}{P_1|P_2 \equiv R_1|R_2} \qquad \textbf{LCom}\ \dfrac{R_1 \xrightarrow{x(y)} Q_1 \qquad R_2 \xrightarrow{\overline{x}z} Q_2}{R_1|R_2 \xrightarrow{\tau} Q_1\{z/y\}|Q_2}}{P_1|P_2 \xrightarrow{\tau} Q_1\{z/y\}|Q_2}$$

Res

$$\textbf{Res}\ \dfrac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\gamma} Q}{P \xrightarrow{\gamma} Q} \qquad z \notin n(\gamma)}{(\nu z)P \xrightarrow{\gamma} (\nu z)Q}$$

become

$$\textbf{Cong}\ \dfrac{\dfrac{P \equiv R}{(\nu z)P \equiv (\nu z)R} \qquad \textbf{Res}\ \dfrac{R \xrightarrow{\gamma} Q \qquad z \notin n(\gamma)}{(\nu z)R \xrightarrow{\gamma} (\nu z)Q}}{(\nu z)P \xrightarrow{\gamma} (\nu z)Q}$$

Opn

$$\textbf{Opn}\ \dfrac{\textbf{Cong}\ \dfrac{P \equiv R \qquad R \xrightarrow{\overline{x}y} Q}{P \xrightarrow{\overline{x}y} Q} \qquad y \neq x}{(\nu y)P \xrightarrow{\overline{x}(y)} Q}$$

become

$$\textbf{Cong}\ \dfrac{\dfrac{P \equiv R}{(\nu y)P \equiv (\nu y)R} \qquad \textbf{Opn}\ \dfrac{R \xrightarrow{\overline{x}y} Q \qquad y \neq x}{(\nu y)R \xrightarrow{\overline{x}(y)} Q}}{(\nu y)P \xrightarrow{\overline{x}(y)} Q}$$

□

Definition 4.3.2. Let $(\nu x)Q$ be an occurrence in a process P, i.e., there is a context $C[\_]$ such that $C[(\nu x)Q] = P$. We say that this occurrence is *guarded* if it occurs right inside a prefix. Otherwise we say that the occurrence is *unguarded*. More formally the occurrence $(\nu x)Q$ is *guarded* in P if there is a context $C[\_]$, an action prefixing α and names $\tilde{y}$ such that $P = C[\alpha.(\nu\tilde{y})(\nu x)Q]$

Definition 4.3.3. We say that a process is in *normal form* if all bound names are distinct and all unguarded restrictions are at the top level, i.e., of the form $(\nu\tilde{x})P$ where P has no unguarded restrictions, note that $\tilde{x}$ can eventually be empty. If a process P is in normal form, we write for short P n.f..

Lemma 4.3.2. Every process is structurally congruent to a process in normal form.

Proof. Let P be a process. We have to show that there exists a process N such that $P \equiv N$ and N is in normal form. We prove this by structural induction on P:

0 in this case $P = 0$ is already in normal form.

$\alpha.P_1$ for inductive hypothesis there exists a process N such that $P_1 \equiv N$ and N is in normal form. Then $\alpha.P_1 \equiv \alpha.N$ and $\alpha.N$ is in normal form.

$P_1 + P_2$ for inductive hypothesis there exist processes N_1 and N_2 such that $P_1 \equiv N_1$, $P_2 \equiv N_2$ and N_1, N_2 are in normal form. If N_1 or N_2 have unguarded restrictions at the top level then $N_1 + N_2$ is not in normal form but we can move the restrictions up to the top level using α equivalence and the rule

$$(\nu x)(P + Q) \equiv P + (\nu x)Q \quad \text{if } x \notin fn(P)$$

and we get something that is in normal form and structurally equivalent to $N_1 + N_2$ and so to $P_1 + P_2$.

$P_1|P_2$ similar.

$(\nu x)P_1$ for inductive hypothesis there exists a process N such that $P_1 \equiv N$ and N is in normal form. $(\nu x)N$ is in normal form and it is structurally congruent to P.

□

Lemma 4.3.3. $P \xrightarrow{\gamma} Q$, $P \equiv N$, N is in normal form then $N \xrightarrow{\gamma} M$, $Q \equiv M$, M is in normal form and the depth of the inference tree of $N \xrightarrow{\gamma} M$ is not greater than the depth of the inference tree of $P \xrightarrow{\gamma} Q$.

Proof. The proof is by induction on the derivation of $P \equiv N$. The last rule used can be:

α *conversion* ?? ???

□

Lemma 4.3.4. If $P \xrightarrow{\gamma} Q$ then there exist processes N, M in normal form such that $P \equiv N$, $N \xrightarrow{\gamma} M$, $Q \equiv M$ and the inference tree of $N \xrightarrow{\gamma} M$ is not deeper than the one of $P \xrightarrow{\gamma} Q$.

Proof. this lemma follows from lemma 4.3.2 and lemma 4.3.3 □

Lemma 4.3.5 (Inversion lemma for structural congruence for normal form)**.**

Proposition 4.3.6. Suppose that we replace the rules *LInp* and *SInp* with the following:

$$\textbf{Inp}\ \frac{n \geq 0}{\underline{x_1(y_1)}.\ \ldots\ .\underline{x_n(y_n)}.z(w).P \xrightarrow{\widetilde{x(y)}\cdot z(w)} P}$$

then the semantic does not change. Also if $P \xrightarrow{\sigma} Q$ then there exist processes N, R such that: $P \equiv N \overset{\sigma}{\twoheadrightarrow} R \equiv M$ and N is in normal form. SARA' VERO?

Proof. The proof is an induction on the depth of $P \xrightarrow{\sigma} Q$. The last rule used can be:

Tau $P = \tau.P_1 \xrightarrow{\tau} P_1 = Q$. For lemma 4.3.2 there exists a normal form N such that $\tau.P_1 \equiv N$. For lemma 4.3.5 $N = \tau.N_1$ and $P_1 \equiv N_1$. So for rule *Tau*: $P \equiv \tau.N_1 \xrightarrow{\tau} N_1 \equiv Q$

Inp $P = \underline{x_1(y_1)}.\ \ldots\ .\underline{x_n(y_n)}.z(w).P_1 \xrightarrow{x_1(y_1)\cdot\ldots\cdot x_n(y_n)\cdot z(w)} P_1 = Q$. For lemma 4.3.2 there exists a normal form N such that $P \equiv N$. For lemma 4.3.5 $N = \underline{x_1(y_1)}.\ \ldots\ .\underline{x_n(y_n)}.z(w).N_1$ and $P_1 \equiv N_1$. For rule *Inp*: $P \equiv \underline{x_1(y_1)}.\ \ldots\ .\underline{x_n(y_n)}.z(w).N_1 \xrightarrow{x_1(y_1)\cdot\ldots\cdot x_n(y_n)\cdot z(w)} N_1 \equiv Q$

Out similar.

Sum $P = P_1 + P_2 \xrightarrow{\gamma} P_1' = Q$. non si puo' applicare l'ipotesi induttiva alle premesse della regola sum.

□

Definition 4.3.4. The *late transition relation for normal forms* is the smallest relation induced by the rules in table 4.4, written $\rightarrow_n$. Every process in the head of transition in the premise of a rule in table 4.4 is assumed to be in normal form. Also when we write $(\nu\tilde{x})P$ is a normal form, it means that P has no restriction at the top level.

Lemma 4.3.7. $P \xrightarrow{\gamma} Q$ imply $P \equiv N \xrightarrow{\gamma}_n M \equiv Q$ for some processes N and M in normal form. Also $N \xrightarrow{\gamma}_n M$ imply $N \xrightarrow{\gamma} M$

$$\textbf{Out}\ \frac{}{\overline{x}y.N \xrightarrow{\overline{x}y}_n N} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau}_n P} \qquad \textbf{Inp}\ \frac{n \geq 0}{\underline{x_1(y_1)}.\ \ldots\ .\underline{x_n(y_n)}.z(w).N \xrightarrow{\widetilde{x(y)}\cdot z(w)}_n N}$$

$$\textbf{LComSeq1}\ \frac{(\nu\tilde{a})P \xrightarrow{x(y)\cdot\sigma}_n (\nu\tilde{b})P^{'} \qquad (\nu\tilde{c})Q \xrightarrow{\overline{x}z}_n (\nu\tilde{d})Q^{'} \qquad bn(\sigma) \cap fn(Q) = \emptyset}{(\nu\tilde{a}\tilde{c})(P|Q) \xrightarrow{\sigma\{z/y\}}_n (\nu\tilde{b}\tilde{d})(P^{'}\{z/y\}|Q^{'})}$$

$$\textbf{LCom1}\ \frac{(\nu\tilde{a})P \xrightarrow{x(y)}_n (\nu\tilde{b})P^{'} \qquad (\nu\tilde{c})Q \xrightarrow{\overline{x}z}_n (\nu\tilde{d})Q^{'}}{(\nu\tilde{a}\tilde{b})(P|Q) \xrightarrow{\tau}_n (\nu\tilde{c}\tilde{d})(P^{'}\{z/y\}|Q^{'})}$$

$$\textbf{LComSeq2}\ \frac{(\nu\tilde{a})P \xrightarrow{\overline{x}z}_n (\nu\tilde{b})P^{'} \qquad (\nu\tilde{c})Q \xrightarrow{x(y)\cdot\sigma}_n (\nu\tilde{d})Q^{'} \qquad bn(\sigma) \cap fn(Q) = \emptyset}{(\nu\tilde{a}\tilde{c})(P|Q) \xrightarrow{\sigma\{z/y\}}_n (\nu\tilde{b}\tilde{d})(P^{'}\{z/y\}|Q^{'})}$$

$$\textbf{LCom2}\ \frac{(\nu\tilde{a})P \xrightarrow{\overline{x}z}_n (\nu\tilde{b})P^{'} \qquad (\nu\tilde{c})Q \xrightarrow{x(y)}_n (\nu\tilde{d})Q^{'}}{(\nu\tilde{a}\tilde{b})(P|Q) \xrightarrow{\tau}_n (\nu\tilde{c}\tilde{d})(P^{'}\{z/y\}|Q^{'})}$$

$$\textbf{Sum1}\ \frac{(\nu\tilde{a})P \xrightarrow{\sigma}_n (\nu\tilde{b})P^{'} \qquad (\nu\tilde{c})Q\ n.\ f.}{(\nu\tilde{a}\tilde{c})(P+Q) \xrightarrow{\sigma}_n (\nu\tilde{b}\tilde{c})P^{'}} \qquad \textbf{Sum2}\ \frac{(\nu\tilde{a})P\ n.\ f. \qquad (\nu\tilde{b})Q \xrightarrow{\sigma}_n (\nu\tilde{c})Q^{'}}{(\nu\tilde{a}\tilde{c})(P+Q) \xrightarrow{\sigma}_n (\nu\tilde{b}\tilde{c})Q^{'}}$$

$$\textbf{Res}\ \frac{(\nu\tilde{a})P \xrightarrow{\sigma}_n (\nu\tilde{b})P^{'} \qquad z \notin n(\alpha)}{(\nu z\tilde{a})P \xrightarrow{\sigma}_n (\nu z\tilde{b})P^{'}} \qquad \textbf{Opn}\ \frac{(\nu\tilde{a})P \xrightarrow{\overline{x}z}_n P^{'} \qquad z \neq x}{(\nu z\tilde{a})P \xrightarrow{\overline{x}(z)}_n P^{'}}$$

$$\textbf{Par1}\ \frac{(\nu\tilde{a})P \xrightarrow{\sigma}_n (\nu\tilde{b})P^{'} \qquad bn(\sigma) \cap fn(Q) = \emptyset \qquad (\nu\tilde{c})Q\ n.\ f.}{(\nu\tilde{a}\tilde{c})(P|Q) \xrightarrow{\sigma}_n (\nu\tilde{b}\tilde{c})(P^{'}|Q)}$$

$$\textbf{Par2}\ \frac{(\nu\tilde{a})P\ n.\ f. \qquad bn(\sigma) \cap fn((\nu\tilde{a})P) = \emptyset \qquad (\nu\tilde{b})Q \xrightarrow{\sigma}_n (\nu\tilde{c})Q^{'}}{(\nu\tilde{a}\tilde{c})(P|Q) \xrightarrow{\sigma}_n (\nu\tilde{b}\tilde{c})(P|Q^{'})}$$

Table 4.4: Multi π late semantic for normal forms. Every process in the head of a transition in the premise of a rule is in normal form. The restrictions can be empty

4.4 Strong bisimilarity and equivalence

4.4.1 Strong bisimilarity

In the following $\widetilde{x(y)} = x_1(y_1) \cdot \ldots \cdot x_n(y_n)$ and $\tilde{x} = x_1 \cdot \ldots \cdot x_n$.

Definition 4.4.1. A *strong bisimulation* is a simmetric binary relation $\mathbf{S}$ on multi π processes such that for all $P\mathbf{S}Q$:

- $P \xrightarrow{\alpha} P'$, $bn(\alpha)$ is fresh and α is not an input nor a sequence of inputs then there exists some Q' such that $Q \xrightarrow{\alpha} Q'$ and $P'\mathbf{S}Q'$
- $P \xrightarrow{\widetilde{x(y)}} P'$ where γ is a possibly empty sequence of inputs and $\tilde{y}$ is fresh then there exists some Q' such that $Q \xrightarrow{\widetilde{x(y)}} Q'$ and for all $\tilde{w}$, $P'\{\tilde{w}/\tilde{y}\}\mathbf{S}Q'\{\tilde{w}/\tilde{y}\}$

P and Q are strongly bisimilar, written $P\dot{\sim}Q$, if they are related by a strong bisimulation.

Is this definition a proper extension of the one in [4]? The only way to tell is by showing some example of process that we intuitively want to be bisimilar.

Example :

$$P = \underline{a(u)}.b(v).0 \quad P\dot{\sim}Q \quad \underline{a(x)}.b(v).(\nu y)\overline{y}u.0 = Q$$

This is because for all $u \in \mathbf{N} - \{b\}$ and for all $v \in \mathbf{N} - \{u\}$: $P \xrightarrow{a(u)\cdot b(v)} 0$. For all $x \in \mathbf{N} - \{b, u\}$ and for all $v \in \mathbf{N} - \{u, x, y\}$:$Q \xrightarrow{a(x)\cdot b(v)} 0$. Taking z, w fresh in P and Q means: $z, w \in \mathbf{N} - \{a, b, u\}$, so both P and Q can make the transition $\xrightarrow{a(z)\cdot b(w)}$ and arrive to 0.

Definition 4.4.2. Let $\mathbf{R}$ be a strong late bisimulation. A *strong bisimulation up to* $\mathbf{R}$ is a simmetric binary relation $\mathbf{S}$ on multi π processes such that for all $P\mathbf{S}Q$:

- $P \xrightarrow{\alpha} P'$, $bn(\alpha)$ is fresh and α is not an input nor a sequence of inputs then there exist processes Q', Q'', P'' such that $Q \xrightarrow{\alpha} Q'$ and $P'\mathbf{R}P''\mathbf{S}Q''\mathbf{R}Q'$
- $P \xrightarrow{x_1(y_1)\cdot\ldots\cdot x_n(y_n)} P'$ where γ is a possibly empty sequence of inputs and $y_1 \cdot \ldots \cdot y_n$ is fresh then there exists some Q' such that $Q \xrightarrow{x_1(y_1)\cdot\ldots\cdot x_n(y_n)} Q'$ and for all $w_1 \cdot \ldots \cdot w_n$ $P'\{w_1/y_1, \ldots, w_n/y_n\}\mathbf{RSR}Q'\{w_1/y_1, \ldots, w_n/y_n\}$

P and Q are strongly bisimilar up to $\mathbf{R}$, written $P\dot{\sim}^{\mathbf{R}}Q$, if they are related by a strong bisimulation up to $\mathbf{R}$.

Proposition 4.4.1. $P\dot{\sim}^{\mathbf{R}}Q$ imply $P\dot{\sim}Q$.

Proof. Let $\mathbf{S}$ be a bisimulation up to $\mathbf{R}$ such that $P\mathbf{S}Q$. It can be proved that $\mathbf{RSR}$ is a bisimulation: let $A\mathbf{R}B\mathbf{S}C\mathbf{R}D$ and let γ be a non input action

$$\begin{array}{l}
A \xrightarrow{\gamma} A' \wedge A\mathbf{R}B \wedge \text{definition 4.4.1} \Rightarrow \exists B' : B \xrightarrow{\gamma} B' \wedge A'\mathbf{R}B' \\
B\mathbf{S}C \wedge \text{definition 4.4.2} \Rightarrow \exists C'C''B'' : C \xrightarrow{\gamma} C' \wedge B'\mathbf{R}B''\mathbf{S}C''\mathbf{R}C' \\
C \xrightarrow{\gamma} C' \wedge C\mathbf{R}D \wedge \text{definition 4.4.1} \Rightarrow \exists D' : D \xrightarrow{\gamma} D' \wedge C'\mathbf{R}D' \\
A'\mathbf{R}B'\mathbf{R}B''\mathbf{S}C''\mathbf{R}C'\mathbf{R}D' \wedge \text{transitivity of } \mathbf{R} \Rightarrow A'\mathbf{R}B''\mathbf{S}C''\mathbf{R}D'
\end{array}$$

It is easy to see that the simmetric also holds. For the other case: let $x_1(y_1) \cdot \ldots \cdot x_n(y_n) = \tilde{x}(\tilde{y})$

$$\begin{array}{l}
A \xrightarrow{\tilde{x}(\tilde{y})} A' \wedge A\mathbf{R}B \wedge \text{definition 4.4.1} \Rightarrow \exists B' : B \xrightarrow{\tilde{x}(\tilde{y})} B' \text{ and for all } \tilde{w} : A'\{\tilde{w}/\tilde{y}\}\mathbf{R}B'\{\tilde{w}/\tilde{y}\} \\
B\mathbf{S}C \wedge \text{definition 4.4.2} \Rightarrow \exists C' : C \xrightarrow{\tilde{x}(\tilde{y})} C' \wedge B'\{\tilde{w}/\tilde{y}\}\mathbf{RSR}C'\{\tilde{w}/\tilde{y}\} \\
C \xrightarrow{\tilde{x}(\tilde{y})} C' \wedge C\mathbf{R}D \wedge \text{definition 4.4.1} \Rightarrow \exists D' : D \xrightarrow{\tilde{x}(\tilde{y})} D' \wedge C'\{\tilde{w}/\tilde{y}\}\mathbf{R}D'\{\tilde{w}/\tilde{y}\} \\
A'\{\tilde{w}/\tilde{y}\}\mathbf{R}B'\{\tilde{w}/\tilde{y}\}\mathbf{RSR}C'\{\tilde{w}/\tilde{y}\}\mathbf{R}D'\{\tilde{w}/\tilde{y}\} \wedge \text{transitivity of } \mathbf{R} \Rightarrow A'\{\tilde{w}/\tilde{y}\}\mathbf{RSR}D'\{\tilde{w}/\tilde{y}\}
\end{array}$$

It is easy to see that the simmetric also holds. $\square$

Proposition 4.4.2. Structural congruence is a strong bisimulation.

Proof. Let $P \equiv Q$. If $P \xrightarrow{\sigma} P^{'}$ then for symmetry of $\equiv$ and rule *Cong*: $Q \xrightarrow{\sigma} P^{'}$. If $Q \xrightarrow{\sigma} Q^{'}$ then for rule *Cong*: $P \xrightarrow{\sigma} Q^{'}$ □

Proposition 4.4.3. $\dot{\sim}$ is preserved by all operators except input prefix.

Proof. We have to try each operator in turn and prove that $\dot{\sim}^{\equiv}$ is preserved:

Output prefix

Let $P\dot{\sim}Q$ and let $\overline{x}y.P \xrightarrow{\alpha} P^{'}$. The last rule used in the derivation of this transition can be:

Out $\overline{x}y.P \xrightarrow{\overline{x}y} P$ and $\overline{x}y.Q \xrightarrow{\overline{x}y} Q$ and $P\dot{\sim}Q$

Cong For lemma **??** a process structurally congruent to $\overline{x}y.P$ must be in the form $\overline{x}y.R$ where $P \equiv R$ so $\overline{x}y.P \xrightarrow{\overline{x}y} R$.

Tau prefix similar.

Input prefix FARE UN ESEMPIO A PARTE DEL PERCH NON FUNZIONA

Strong input FARE UN ESEMPIO A PARTE DEL PERCH NON FUNZIONA

Summation QUESTA DIMOSTRAZIONE NON FUNZIONA PERCHE' IL LEMMA 4.3.1 E' FALSO!!!

Let $P\dot{\sim}Q$ and let $P + R \xrightarrow{\gamma} P^{'}$. The last rule used in the derivation of this transition can be:

Sum $P + R \xrightarrow{\gamma} P^{'}$ because $P \xrightarrow{\gamma} P^{'}$ so $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\dot{\sim}Q^{'}$ or $P^{'}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{'}\{\tilde{w}/\tilde{y}\}$

Cong For proposition 4.3.1 we can assume that only the last rule used to prove $P + R \xrightarrow{\gamma} P^{'}$ is *Cong* so

$$\textbf{Cong}\ \frac{P + R \equiv S \qquad S \overset{\gamma}{\twoheadrightarrow} P^{'}}{P + R \xrightarrow{\gamma} P^{'}}$$

we proceed by cases on the last rule used in the derivation of $P + R \equiv S$:

*Cong*2 S is $A+B$, $P \equiv A$ and $R \equiv B$. Then $A+B \overset{\gamma}{\twoheadrightarrow} P^{'}$, the last rule used in this derivation must be *Sum* so $A \overset{\gamma}{\twoheadrightarrow} P^{'}$.

$$\textbf{Cong}\ \frac{P \equiv A \qquad A \overset{\gamma}{\twoheadrightarrow} P^{'}}{P \xrightarrow{\gamma} P^{'}}$$

Since $P\dot{\sim}Q$ we have $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\dot{\sim}Q^{'}$ or $P^{'}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{'}\{\tilde{w}/\tilde{y}\}$. For rule *Sum*: $Q+R \xrightarrow{\gamma} Q^{'}$

SumCom S is $R + P$. Then $R + P \overset{\gamma}{\twoheadrightarrow} P^{'}$, the last rule used in this derivation must be *Sum* so $R \overset{\gamma}{\twoheadrightarrow} P^{'}$.

$$\textbf{Cong}\ \frac{Q + R \equiv R + Q \qquad \textbf{Sum}\ \dfrac{R \overset{\gamma}{\twoheadrightarrow} P^{'}}{R + Q \xrightarrow{\gamma} P^{'}}}{Q + R \xrightarrow{\gamma} P^{'}}$$

Alp S is α equivalent to $P + R$ so $S = S_1 + S_2$ such that $S_1 \equiv_\alpha P$ and $S_2 \equiv_\alpha R$. Then $S_1 + S_2 \overset{\gamma}{\twoheadrightarrow} P^{'}$, the last rule used in this derivation must be *Sum* so $S_1 \overset{\gamma}{\twoheadrightarrow} P^{'}$.

$$\textbf{Cong}\ \frac{P \equiv_\alpha S_1 \qquad S_1 \overset{\gamma}{\twoheadrightarrow} P^{'}}{P \xrightarrow{\gamma} P^{'}}$$

Since $P\dot{\sim}Q$ we have $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\dot{\sim}Q^{'}$ or $P^{'}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{'}\{\tilde{w}/\tilde{y}\}$. For rule *Sum*: $Q+R \xrightarrow{\gamma} Q^{'}$

ScpExtSum2

$$\textbf{Cong}\ \frac{\dfrac{x \notin fn(P)}{P+(\nu x)R \equiv (\nu x)(P+R)} \qquad (\nu x)(P+R) \overset{\gamma}{\twoheadrightarrow} P^{'}}{P+R \xrightarrow{\gamma} P^{'}}$$

the last rule used in the derivation of $(\nu x)(P+R) \overset{\gamma}{\twoheadrightarrow} P^{'}$ can be:

Res

$$\textbf{Res}\ \frac{\textbf{Sum}\ \dfrac{P \overset{\gamma}{\twoheadrightarrow} P^{''}}{P+R \overset{\gamma}{\twoheadrightarrow} P^{''}} \qquad x \notin n(\gamma)}{(\nu x)(P+R) \overset{\gamma}{\twoheadrightarrow} (\nu x)P^{''}}$$

$P\dot{\sim}Q$ and $P \overset{\gamma}{\twoheadrightarrow} P^{''}$ imply $Q \xrightarrow{\gamma} Q^{''}$ and $P^{''}\dot{\sim}Q^{''}$ or $P^{''}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{''}\{\tilde{w}/\tilde{y}\}$. For rules *Res* and *Sum*: $Q+(\nu x)R \xrightarrow{\gamma} (\nu x)Q^{''}$.

Opn

$SumAsc1(1)$

$$\textbf{Cong}\ \frac{\textbf{SumAsc1}\ \dfrac{}{(P_1+P_2)+R \equiv P_1+(P_2+R)} \qquad \textbf{Sum}\ \dfrac{P_1 \overset{\gamma}{\twoheadrightarrow} P^{'}}{P_1+(P_2+R) \overset{\gamma}{\twoheadrightarrow} P^{'}}}{(P_1+P_2)+R \xrightarrow{\gamma} P^{'}}$$

$P_1 \overset{\gamma}{\twoheadrightarrow} P^{'}$ imply $P = P_1+P_2 \overset{\gamma}{\twoheadrightarrow} P^{'}$ so for bisimilarity $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\dot{\sim}Q^{'}$ or $P^{'}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{'}\{\tilde{w}/\tilde{y}\}$. For rule *Sum*: $Q+R \xrightarrow{\gamma} Q^{'}$.

$SumAsc1(2)$

$$\textbf{Cong}\ \frac{\textbf{SumAsc1}\ \dfrac{}{(P+R_1)+R_2 \equiv P+(R_1+R_2)} \qquad \textbf{Sum}\ \dfrac{P \overset{\gamma}{\twoheadrightarrow} P^{'}}{P+(R_1+R_2) \overset{\gamma}{\twoheadrightarrow} P^{'}}}{(P+R_1)+R_2 \xrightarrow{\gamma} P^{'}}$$

$P \overset{\gamma}{\twoheadrightarrow} P^{'}$ so for bisimilarity $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\dot{\sim}Q^{'}$ or $P^{'}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{'}\{\tilde{w}/\tilde{y}\}$. For rule *Sum*: $Q+R \xrightarrow{\gamma} Q^{'}$.

$SumAsc2(2)$

$$\textbf{Cong}\ \frac{P_1+(P_2+R) \equiv (P_1+P_2)+R \qquad \textbf{Sum}\ \dfrac{\textbf{Sum}\ \dfrac{P_1 \overset{\gamma}{\twoheadrightarrow} P^{'}}{P = P_1+P_2 \xrightarrow{\gamma} P^{'}}}{(P_1+P_2)+R \xrightarrow{\gamma} P^{'}}}{P_1+(P_2+R) \xrightarrow{\gamma} P^{'}}$$

$P \overset{\gamma}{\twoheadrightarrow} P^{'}$ so for bisimilarity $Q \xrightarrow{\gamma} Q^{'}$ and $P^{'}\dot{\sim}Q^{'}$ or $P^{'}\{\tilde{w}/\tilde{y}\}\dot{\sim}Q^{'}\{\tilde{w}/\tilde{y}\}$. For rule *Sum*: $Q+R \xrightarrow{\gamma} Q^{'}$.

Restriction

The relation

$$Res(\dot{\sim}) = \{((\nu x)P, (\nu x)Q) : P\dot{\sim}Q\} \cup \dot{\sim}$$

is a strong bisimulation. There are some cases to consider depending on rule applicable to $(\nu x)P$:

$Res(1)$ let $\tilde{y}$ be fresh in P, Q.

$$\textbf{Res}\ \frac{P \xrightarrow{\widetilde{x(y)}} P' \qquad z \notin n(\widetilde{x(y)})}{(\nu z)P \xrightarrow{\widetilde{x(y)}} (\nu z)P'}$$

$P \xrightarrow{\widetilde{x(y)}} P'$ and $P \dot{\sim} Q$ imply $Q \xrightarrow{\widetilde{x(y)}} Q'$ and for all $\tilde{w}$: $P'\{\tilde{w}/\tilde{y}\} \dot{\sim} Q'\{\tilde{w}/\tilde{y}\}$ which imply $(\nu z)(P'\{\tilde{w}/\tilde{y}\}) Res(\dot{\sim})(\nu z)(Q'\{\tilde{w}/\tilde{y}\})$. Under the hypothesis that $z \notin \tilde{w}$: $z \notin n(\widetilde{x(y)})$ imply $(\nu z)(P'\{\tilde{w}/\tilde{y}\}) = ((\nu z)P')\{\tilde{w}/\tilde{y}\}$. Nevertheless we have to prove that also for $z \in \tilde{w}$ and $z \notin n(\widetilde{x(y)})$: $((\nu z)P')\{\tilde{w}/\tilde{y}\} Res(\dot{\sim})((\nu z)Q')\{\tilde{w}/\tilde{y}\}$. COME ?!?!?!?!

$Res(2)$ let γ be a non input action

$$\textbf{Res}\ \frac{P \xrightarrow{\gamma} P' \qquad z \notin n(\gamma)}{(\nu z)P \xrightarrow{\gamma} (\nu z)P'}$$

$P \xrightarrow{\gamma} P'$ and $P \dot{\sim} Q$ imply $Q \xrightarrow{\gamma} Q'$ and $P' \dot{\sim} Q'$ which in turn imply $(\nu z)P' Res(\dot{\sim})(\nu z)Q'$.

Opn let $\tilde{y}$ be fresh in P, Q.

$$\textbf{Opn}\ \frac{P \xrightarrow{\overline{x}y} P'}{(\nu y)P \xrightarrow{\overline{x}(y)} P'}$$

$P \xrightarrow{\overline{x}y} P'$ and $P \dot{\sim} Q$ imply $Q \xrightarrow{\overline{x}y} Q'$ and $P' \dot{\sim} Q'$ which imply that $((\nu z)P', (\nu z)Q')$ is in $Res(\dot{\sim})$.

$Cong \rightarrow_n$ for lemma 4.3.7 we can assume that the proof tree of $(\nu x)P \xrightarrow{\widetilde{x(y)}} P'$ ends in the following way:

$$\textbf{Cong}\ \frac{(\nu z)P \equiv R \qquad R \xrightarrow{\widetilde{x(y)}}_n P'}{(\nu z)P \xrightarrow{\widetilde{x(y)}} P'}$$

where R is in normal form. At this point the last rule of a derivation of $R \xrightarrow{\widetilde{x(y)}}_n P'$ can be:

Inp this case does not exist because $(\nu a)B \not\equiv c(d).E$

$LComSeq$

$$\textbf{LComSeq1}\ \frac{(\nu\tilde{a})R_1 \xrightarrow{x(y)\cdot\sigma}_n (\nu\tilde{b})R_1' \qquad (\nu\tilde{c})R_2 \xrightarrow{\overline{x}z}_n (\nu\tilde{d})R_2' \qquad bn(\sigma) \cap fn(Q) = \emptyset}{(\nu\tilde{a}\tilde{c})(R_1|R_2) \xrightarrow{\sigma\{z/y\}}_n (\nu\tilde{b}\tilde{d})(R_1'|R_2')}$$

$(\nu z)P \equiv (\nu\tilde{a}\tilde{c})(R_1|R_2)$ and $\sigma\{z/y\} = \widetilde{x(y)}$.

$Sum1, 2$

Res

$Par1, 2$

$Cong \twoheadrightarrow$ for lemma 4.3.1 we can assume that the proof tree of $(\nu x)P \xrightarrow{\widetilde{x(y)}} P'$ ends in the following way:

$$\textbf{Cong}\ \frac{(\nu z)P \equiv R \qquad R \overset{\widetilde{x(y)}}{\twoheadrightarrow} P'}{(\nu z)P \xrightarrow{\widetilde{x(y)}} P'}$$

so the proof goes on by cases on the last rule of the inference of $(\nu z)P \equiv R$ which bearing in mind lemma RICONTROLLARE LA DIMOSTRAZIONE PERCHE' IL LEMMA ERA FALSO! can be:

ResCom so arranging some names in order to make it look more clear, the last part of the inference is:

$$\textbf{Cong}\ \frac{\textbf{ResCom}\ \dfrac{}{(\nu z)(\nu w)P \equiv (\nu w)(\nu z)P} \qquad \textbf{Res}\ \dfrac{\textbf{Res}\ \dfrac{P \overset{\widetilde{x(y)}}{\twoheadrightarrow} P' \qquad w, z \notin n(\widetilde{x(y)})}{(\nu z)P \overset{\widetilde{x(y)}}{\twoheadrightarrow} (\nu z)P'}}{(\nu w)(\nu z)P \overset{\widetilde{x(y)}}{\twoheadrightarrow} (\nu w)(\nu z)P'}}{(\nu z)(\nu w)P \xrightarrow{\widetilde{x(y)}} P'}$$

Trans

ScpExtPar1 $\textbf{ScpExtPar1}\ \dfrac{z \notin fn(P_1)}{(\nu z)(P_1|P_2) \equiv P_1|(\nu z)P_2}$

ScpExtSum1 $\textbf{ScpExtSum1}\ \dfrac{z \notin fn(P_1)}{(\nu z)(P_1 + P_2) \equiv P_1 + (\nu z)P_2}$

Alp $\textbf{Alp}\ \dfrac{P \equiv_\alpha Q}{P \equiv Q}$

Parallel *SumAsc*1 **SumAsc1** $M_1 + (M_2 + M_3) \equiv (M_1 + M_2) + M_3$

*ParAsc*1 **ParAsc1** $P_1|(P_2|P_3) \equiv (P_1|P_2)|P_3$

*SumAsc*2 **SumAsc2** $(M_1 + M_2) + M_3 \equiv M_1 + (M_2 + M_3)$

*ParAsc*2 **ParAsc2** $(P_1|P_2)|P_3 \equiv P_1|(P_2|P_3)$

ParCom **ParCom** $P_1|P_2 \equiv P_2|P_1$

ResCom **ResCom** $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$

SumCom **SumCom** $M_1 + M_2 \equiv M_2 + M_1$

ScpExtPar1 $\textbf{ScpExtPar1}\ \dfrac{z \notin fn(P_1)}{(\nu z)(P_1|P_2) \equiv P_1|(\nu z)P_2}$

ScpExtPar2 $\textbf{ScpExtPar2}\ \dfrac{z \notin fn(P_1)}{P_1|(\nu z)P_2 \equiv (\nu z)(P_1|P_2)}$

ScpExtSum1 $\textbf{ScpExtSum1}\ \dfrac{z \notin fn(P_1)}{(\nu z)(P_1 + P_2) \equiv P_1 + (\nu z)P_2}$

ScpExtSum2 $\textbf{ScpExtSum2}\ \dfrac{z \notin fn(P_1)}{P_1 + (\nu z)P_2 \equiv (\nu z)(P_1 + P_2)}$

Ide $\textbf{Ide}\ \dfrac{A(\tilde{x}) \overset{def}{=} P}{A(\tilde{w}) \equiv P\{\tilde{w}/\tilde{x}\}}$

Trans $\textbf{Trans}\ \dfrac{P \equiv Q \qquad Q \equiv R}{P \equiv R}$

Alp $\textbf{Alp}\ \dfrac{P \equiv_\alpha Q}{P \equiv Q}$

$$Cong1 \quad \textbf{Cong1}\ \frac{P \equiv Q}{C[P] \equiv C[Q]}$$

$$Cong2 \quad \textbf{Cong2}\ \frac{P_1 \equiv Q_1 \qquad P_2 \equiv Q_2}{C[P_1, P_2] \equiv C[Q_1, Q_2]}$$

$\square$

Chapter 5

Multi π calculus with strong input and output

5.1 Syntax

As we did with multi π calculus, we suppose that we have a countable set of names $\mathbb{N}$, ranged over by lower case letters $a, b, \cdots, z$. This names are used for communication channels and values. Furthermore we have a set of identifiers, ranged over by A. We represent the agents or processes by upper case letters $P, Q, \cdots$. A multi π process, in addiction to the same actions of a π process, can perform also a strong prefix:

$$\pi ::= \overline{x}y \mid x(z) \mid \underline{x(y)} \mid \underline{\overline{x}y} \mid \tau$$

The process are defined, just as original π calculus, by the following grammar:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P + Q \mid (\nu x)P \mid A(y_1, \cdots, y_n)$$

and they have the same intuitive meaning as for the π calculus. The strong prefix input allows a process to make an atomic sequence of actions, so that more than one process can synchronize on this sequence.

We have to extend the following definition to deal with the strong prefix:

$$\begin{array}{ll} B(\underline{x(y)}.Q, I) = \{y, \overline{y}\} \cup B(Q, I) & F(\underline{x(y)}.Q, I) = \{x, \overline{x}\} \cup (F(Q, I) - \{y, \overline{y}\}) \\ B(\underline{\overline{x}y}.Q, I) = B(Q, I) & F(\underline{\overline{x}y}.Q, I) = \{x, \overline{x}, y, \overline{y}\} \cup F(Q, I) \end{array}$$

5.2 Operational semantic

5.2.1 Early operational semantic with structural congruence

Definition 5.2.1. The *early transition relation with structural congruence* is the smallest relation induced by the rules in table 5.1:

The names $\sigma, \sigma_1, \sigma_2, \sigma_3$ are non empty sequences of actions and are also not τ. The relation *ESync* is defined by the axioms in table 5.2

Example *Transactional synchronization.* This is an example of two processes that synchronize over a sequence of actions of length two:

$$\underline{\overline{a}x}.\overline{a}y.P|\underline{a(w)}.a(z).Q \xrightarrow{\tau} P|Q\{x/w\}\{y/z\}$$

We start first noticing that

$$\text{S4R}\ \frac{\text{S1R}\ \dfrac{}{Sync(\overline{a}y, ay, \tau)}}{Sync(\overline{a}x \cdot \overline{a}y, ax \cdot ay, \tau)}$$

and that

$$\textbf{Inp}\ \frac{}{x(y).P \xrightarrow{xz} P\{z/x\}} \qquad \textbf{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textbf{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P}$$

$$\textbf{SInp}\ \frac{P\{z/y\} \xrightarrow{\sigma} P^{'} \qquad \sigma \neq \tau}{\underline{x(y)}.P \xrightarrow{xz\cdot\sigma} P^{'}} \qquad \textbf{SOut}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad \sigma \neq \tau}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\sigma} P^{'}}$$

$$\textbf{ECom}\ \frac{P \xrightarrow{\sigma_1} P^{'} \qquad Q \xrightarrow{\sigma_2} Q^{'} \qquad ESync(\sigma_1, \sigma_2, \sigma_3)}{P|Q \xrightarrow{\sigma_3} P^{'}|Q^{'}}$$

$$\textbf{Sum}\ \frac{P \xrightarrow{\sigma} P^{'}}{P + Q \xrightarrow{\sigma} P^{'}} \qquad \textbf{Cong}\ \frac{P \equiv P^{'} \qquad P^{'} \xrightarrow{\alpha} Q}{P \xrightarrow{\alpha} Q}$$

$$\textbf{Res}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\sigma} (\nu z)P^{'}} \qquad \textbf{Par}\ \frac{P \xrightarrow{\sigma} P^{'}}{P|Q \xrightarrow{\sigma} P^{'}|Q}$$

Table 5.1: Multi π early semantic with structural congruence

$$\text{S1L}\ \frac{}{ESync(xy, \overline{x}y, \tau)} \qquad \text{S1R}\ \frac{}{ESync(\overline{x}y, xy, \tau)}$$

$$\text{S2L}\ \frac{}{ESync(xy, \overline{x}y\cdot\sigma, \sigma)} \qquad \text{S2R}\ \frac{}{ESync(\overline{x}y\cdot\sigma, xy, \sigma)}$$

$$\text{S3L}\ \frac{}{ESync(xy\cdot\sigma, \overline{x}y, \sigma)} \qquad \text{S3R}\ \frac{}{ESync(\overline{x}y, xy\cdot\sigma, \sigma)}$$

$$\text{S4L}\ \frac{ESync(\sigma_1, \sigma_2, \sigma_3)}{ESync(xy\cdot\sigma_1, \overline{x}y\cdot\sigma_2, \sigma_3)} \qquad \text{S4R}\ \frac{ESync(\sigma_1, \sigma_2, \sigma_3)}{ESync(\overline{x}y\cdot\sigma_1, xy\cdot\sigma_2, \sigma_3)}$$

Table 5.2: Synchronization relation

$$\textsc{SOut}\ \dfrac{\textsc{Out}\ \dfrac{}{\overline{a}y.P \xrightarrow{\overline{a}y} P}}{\underline{\overline{a}x}.\overline{a}y.P \xrightarrow{\overline{a}x\cdot\overline{a}y} P} \qquad \textsc{SInp}\ \dfrac{\textsc{Inp}\ \dfrac{}{(a(z).Q)\{x/w\} \xrightarrow{ay} Q\{x/w\}\{y/z\}}}{\underline{a(w)}.a(z).Q \xrightarrow{ax\cdot ay} Q}$$

and in the end we just need to apply the rule **LCom**

Example *Multi-party synchronization.* In this example we have three processes that want to synchronize:

$$\textbf{ECom}\ \dfrac{\underline{\overline{a}f}.\overline{b}g.P|a(w).Q \xrightarrow{\overline{b}g} P|Q\{f/w\} \qquad \textbf{Inp}\ \dfrac{}{b(y).R \xrightarrow{bg} R\{g/y\}} \qquad \textbf{S1R}\ \dfrac{}{Sync(\overline{b}g, bg, \tau)}}{(\underline{\overline{a}f}.\overline{b}g.P|a(w).Q)|b(y).R \xrightarrow{\tau} (P|Q\{f/w\})|R\{g/y\}}$$

$$\textbf{LCom}\ \dfrac{\underline{\overline{a}f}.\overline{b}g.P \xrightarrow{\overline{a}f\cdot\overline{b}g} P \qquad \textbf{Inp}\ \dfrac{}{a(w).Q \xrightarrow{af} Q\{f/w\}} \qquad \textbf{S2R}\ \dfrac{}{Sync(\overline{a}f\cdot\overline{b}g, af, \overline{b}g)}}{\underline{\overline{a}f}.\overline{b}g.P|a(w).Q \xrightarrow{\overline{b}g} P|Q\{f/w\}}$$

$$\textbf{SOut}\ \dfrac{\textbf{Out}\ \dfrac{}{\overline{b}g.P \xrightarrow{\overline{b}g} P}}{\underline{\overline{a}f}.\overline{b}g.P \xrightarrow{\overline{a}f\cdot\overline{b}g} P}$$

5.2.2 Late operational semantic with structural congruence

The semantic of a multi π process is labeled transition system such that

- the nodes are multi π calculus process. The set of node is $\mathbb{P}_m$
- The set of actions is $\mathbb{A}_m$ and can contain
  - bound output $\overline{x}(y)$
  - unbound output $\overline{x}y$
  - bound input $x(z)$

  We use $\alpha, \alpha_1, \alpha_2, \cdots$ to range over the set of actions, we use $\sigma, \sigma_1, \sigma_2, \cdots$ to range over the set $\mathbb{A}_m^+ \cup \{\tau\}$.
- the transition relations is $\rightarrow \subseteq \mathbb{P}_m \times (\mathbb{A}_m^+ \cup \{\tau\}) \times \mathbb{P}_m$

In this case, a label can be a sequence of prefixes, whether in the original π calculus a label can be only a prefix. We use the symbol $\cdot$ to denote the concatenation operator.

Definition 5.2.2. The *late transition relation with structural congruence* is the smallest relation induced by the rules in table 5.3:

In what follows, the names $\delta, \delta_1, \delta_2$ represents substitutions, they can also be empty; the names $\sigma, \sigma_1, \sigma_2, \sigma_3$ are non empty sequences of actions. The relation *Sync* is defined by the axioms in table 5.4

Example *Transactional synchronization.* This is an example of two processes that synchronize over a sequence of actions of length two:

$$\underline{\overline{a}x}.\overline{a}y.P|\underline{a(w)}.a(z).Q \xrightarrow{\tau} P|Q\{x/w\}\{y/z\}$$

We start first noticing that

$$\text{S4R}\ \dfrac{\text{S1R}\ \dfrac{}{Sync(\overline{a}y, a(z)\{x/w\}, \tau, \{\}, \{y/z\})}}{Sync(\overline{a}x\cdot\overline{a}y, a(w)\cdot a(z), \tau, \{\}, \{x/w\}\{y/z\})}$$

and that

$$\textbf{Pref}\ \frac{\alpha\ not\ a\ strong\ prefix}{\alpha.P \xrightarrow{\alpha} P} \qquad \textbf{Par}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad bn(\sigma) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\sigma} P^{'}|Q}$$

$$\textbf{SOut}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad \sigma \neq \tau}{\underline{\overline{x}y}.P \xrightarrow{\overline{x}y\cdot\sigma} P^{'}} \qquad \textbf{LCom}\ \frac{P \xrightarrow{\sigma_1} P^{'} \qquad Q \xrightarrow{\sigma_2} Q^{'} \qquad Sync(\sigma_1, \sigma_2, \sigma_3, \delta_1, \delta_2)}{P|Q \xrightarrow{\sigma_3} P^{'}\delta_1|Q^{'}\delta_2}$$

$$\textbf{Sum}\ \frac{P \xrightarrow{\sigma} P^{'}}{P + Q \xrightarrow{\sigma} P^{'}} \qquad \textbf{Cong}\ \frac{P \equiv P^{'} \qquad P^{'} \xrightarrow{\alpha} Q}{P \xrightarrow{\alpha} Q}$$

$$\textbf{Res}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad z \notin n(\alpha)}{(\nu z)P \xrightarrow{\sigma} (\nu z)P^{'}} \qquad \textbf{SInp}\ \frac{P \xrightarrow{\sigma} P^{'} \qquad \sigma \neq \tau}{\underline{x(y)}.P \xrightarrow{x(y)\cdot\sigma} P^{'}}$$

Table 5.3: Multi π late semantic with structural congruence

$$\text{S1L}\ \frac{}{Sync(x(y), \overline{x}z, \tau, \{z/y\}, \{\})} \qquad \text{S1R}\ \frac{}{Sync(\overline{x}z, x(y), \tau, \{\}, \{z/y\})}$$

$$\text{S2L}\ \frac{}{Sync(x(y), \overline{x}z \cdot \sigma, \sigma, \{z/y\}, \{\})} \qquad \text{S2R}\ \frac{}{Sync(\overline{x}z \cdot \sigma, x(y), \sigma, \{\}, \{z/y\})}$$

$$\text{S3L}\ \frac{}{Sync(x(y) \cdot \sigma, \overline{x}z, \sigma\{z/y\}, \{z/y\}, \{\})} \qquad \text{S3R}\ \frac{}{Sync(\overline{x}z, x(y) \cdot \sigma, \sigma\{z/y\}, \{\}, \{z/y\})}$$

$$\text{S4L}\ \frac{Sync(\sigma_1, \sigma_2\{z/y\}, \sigma_3, \delta_1, \delta_2)}{Sync(x(y) \cdot \sigma_1, \overline{x}z \cdot \sigma_2, \sigma_3, \{z/y\}\delta_1, \delta_2)} \qquad \text{S4R}\ \frac{Sync(\sigma_1, \sigma_2\{z/y\}, \sigma_3, \delta_1, \delta_2)}{Sync(\overline{x}z \cdot \sigma_1, x(y) \cdot \sigma_2, \sigma_3, \delta_1, \{z/y\}\delta_2)}$$

Table 5.4: Synchronization relation

$$\textsc{SOut}\ \dfrac{\textsc{Pref}\ \dfrac{}{\overline{a}y.P \xrightarrow{\overline{a}y} P}}{\underline{\overline{a}x}.\overline{a}y.P \xrightarrow{\overline{a}x\cdot\overline{a}y} P} \qquad \textsc{SInp}\ \dfrac{\textsc{Pref}\ \dfrac{}{a(z).Q \xrightarrow{a(z)} Q}}{\underline{a(w)}.a(z).Q \xrightarrow{a(w)\cdot a(z)} Q}$$

and in the end we just need to apply the rule **LCom**

Example *Multi-party synchronization.* In this example we have three processes that want to synchronize:

$$\textbf{LCom}\ \dfrac{\underline{\overline{a}f}.\overline{b}g.P|a(w).Q \xrightarrow{\overline{b}g} P|Q\{f/w\} \qquad \textbf{Pref}\ \dfrac{}{b(y).R \xrightarrow{b(y)} R} \qquad \textbf{S1R}\ \dfrac{}{Sync(\overline{b}g, b(y), \tau, \emptyset, \{g/y\})}}{(\underline{\overline{a}f}.\overline{b}g.P|a(w).Q)|b(y).R \xrightarrow{\tau} (P|Q\{f/w\})|R\{g/y\}}$$

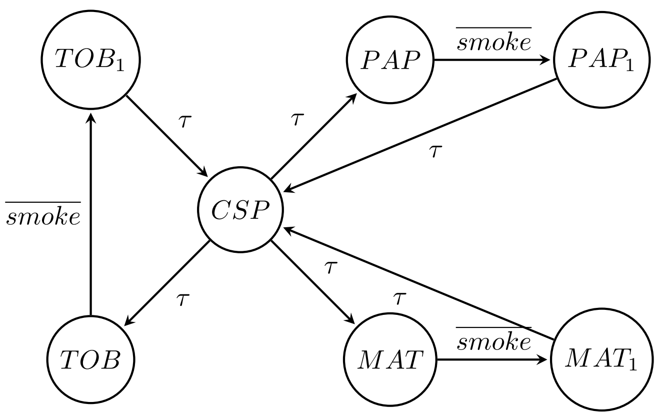
$$\textbf{LCom}\ \dfrac{\underline{\overline{a}f}.\overline{b}g.P \xrightarrow{\overline{a}f\cdot\overline{b}g} P \qquad \textbf{Pref}\ \dfrac{}{a(w).Q \xrightarrow{a(w)} Q} \qquad \textbf{S2R}\ \dfrac{}{Sync(\overline{a}f\cdot\overline{b}g, a(w), \overline{b}g, \emptyset, \{f/w\})}}{\underline{\overline{a}f}.\overline{b}g.P|a(w).Q \xrightarrow{\overline{b}g} P|Q\{f/w\}}$$

$$\textbf{SOut}\ \dfrac{\textbf{Out}\ \dfrac{}{\overline{b}g.P \xrightarrow{\overline{b}g} P}}{\underline{\overline{a}f}.\overline{b}g.P \xrightarrow{\overline{a}f\cdot\overline{b}g} P}$$

Example *Cigarette smokers' problem.* In this problem there are four processes: an agent and three smokers. Each smoker continuously makes a cigarette and smokes it. To make a cigarette each smoker needs three ingredients: tobacco, paper and matches. One of the smokers has paper, another tobacco and the third matches. The agent has an infinite supply of the ingredients. The agent places two of the ingredients on the table. The smoker who has the remaining ingredient take the others from the table, make a cigarette and smokes. Then the cycle repeats. A solution to the problem is the following:

$$\begin{aligned}
&Agent \overset{def}{=} \underline{\overline{tob}}.\overline{mat}.end().Agent + \underline{\overline{mat}}.\overline{pap}.end().Agent + \underline{\overline{pap}}.\overline{tob}.end().Agent\\
&S_{pap} \overset{def}{=} \underline{tob()}.mat().\overline{smoke}.\overline{end}.S_{pap}\\
&S_{tab} \overset{def}{=} \underline{mat()}.pap().\overline{smoke}.\overline{end}.S_{tab}\\
&S_{mat} \overset{def}{=} \underline{pap()}.tob().\overline{smoke}.\overline{end}.S_{mat}\\
&CSP \overset{def}{=} (\nu tob, pap, mat, end)(Agent|S_{tob}|S_{mat}|S_{pap})
\end{aligned}$$

The semantic of CSP is the following graph:



where

$$
\begin{aligned}
&PAP \overset{def}{=} (\nu tob, pap, mat, end)(end().Agent|S_{tob}|S_{mat}|\overline{smoke}.\overline{end}.S_{pap})\\
&TOB \overset{def}{=} (\nu tob, pap, mat, end)(end().Agent|\overline{smoke}.\overline{end}.S_{tob}|S_{mat}|S_{pap})\\
&MAT \overset{def}{=} (\nu tob, pap, mat, end)(end().Agent|S_{tob}|\overline{smoke}.\overline{end}.S_{mat}|S_{pap})\\
&PAP_1 \overset{def}{=} (\nu tob, pap, mat, end)(end().Agent|S_{tob}|S_{mat}|\overline{end}.S_{pap})\\
&TOB_1 \overset{def}{=} (\nu tob, pap, mat, end)(end().Agent|\overline{end}.S_{tob}|S_{mat}|S_{pap})\\
&MAT_1 \overset{def}{=} (\nu tob, pap, mat, end)(end().Agent|S_{tob}|\overline{end}.S_{mat}|S_{pap})
\end{aligned}
$$

5.2.3 Low level semantic

This section contains the definition of an alternative semantic for multi π. First we define a low level version of the multi π calculus, we call this language low multi π. The low multi π is the multi π enriched with a marked or intermediate process $*P$:

$$P, Q ::= 0 \mid \pi.P \mid P|Q \mid P + Q \mid (\nu x)P \mid A(x_1, \cdots, x_n) \mid *P$$

$$\pi ::= \overline{x}y \mid x(y) \mid \underline{\overline{x}y} \mid \underline{x(y)} \mid \tau$$

Definition 5.2.3. The low level transition relation is the smallest relation induced by the rules in table 5.5 in which P stands for a process without mark, L stands for a process with mark and S can stand for both.

Proposition 5.2.1. Let $\rightarrow$ be the relation defined in table 5.1. If $P \xrightarrow{\sigma} Q$ then there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

Proof. The proof is by induction on the depth of the derivation tree of $P \xrightarrow{\sigma} Q$:

base case

If the depth is one then the rule used have to be one of: $EInp$, Out, Tau. These rules are also in table 5.5 so we can derive $P \overset{\sigma}{\longmapsto} Q$.

inductive case

If the depth is greater than one then the last rule used in the derivation can be:

$SOut$: the last part of the derivation tree looks like this:

$$\textbf{SOut}\ \frac{P_1 \xrightarrow{\sigma} Q \qquad \sigma \neq \tau}{\underline{\overline{x}y}.P_1 \xrightarrow{\overline{x}y\cdot\sigma} Q}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \overset{\gamma_1}{\longmapsto} L_1 \overset{\gamma_2}{\longmapsto} L_2 \cdots L_{k-1} \overset{\gamma_k}{\longmapsto} L_k \overset{\gamma_{k+1}}{\longmapsto} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

then a proof of the conclusion follows from:

$$\textbf{SOutLow}\ \frac{}{\underline{\overline{x}y}.P_1 \overset{\overline{x}y}{\longmapsto} *P_1} \qquad \textbf{Star}\ \frac{P_1 \overset{\gamma_1}{\longmapsto} L_1}{*P_1 \overset{\gamma_1}{\longmapsto} L_1}$$

$SInp$: this case is similar to the previous.

Sum : the last part of the derivation tree looks like this:

$$\textbf{Sum}\ \frac{P_1 \xrightarrow{\sigma} Q}{P_1 + P_2 \xrightarrow{\sigma} Q}$$

$$\textbf{Out}\ \frac{}{\overline{x}y.P \xmapsto{\overline{x}y} P} \quad \textbf{EInp}\ \frac{}{x(y).P \xmapsto{xz} P\{z/y\}} \quad \textbf{Tau}\ \frac{}{\tau.P \xmapsto{\tau} P}$$

$$\textbf{SOutLow}\ \frac{}{\underline{\overline{x}y}.P \xmapsto{\overline{x}y} *P} \quad \textbf{SInpLow}\ \frac{}{\underline{x(y)}.P \xmapsto{xz} *P\{z/y\}}$$

$$\textbf{StarEps}\ \frac{S \xmapsto{\epsilon} S^{'}}{*S \xmapsto{\epsilon} S^{'}} \quad \textbf{StarInp}\ \frac{S \xmapsto{xy} S^{'}}{*S \xmapsto{xy} S^{'}} \quad \textbf{StarOut}\ \frac{S \xmapsto{\overline{x}y} S^{'}}{*S \xmapsto{\overline{x}y} S^{'}}$$

$$\textbf{Par1R}\ \frac{S \xmapsto{\gamma} S^{'}}{Q|S \xmapsto{\gamma} Q|S^{'}} \quad \textbf{Par1L}\ \frac{S \xmapsto{\gamma} S^{'}}{S|Q \xmapsto{\gamma} S^{'}|Q}$$

$$\textbf{Sum}\ \frac{P \xmapsto{\gamma} S}{P+Q \xmapsto{\gamma} S} \quad \textbf{Cong}\ \frac{P \equiv P^{'} \qquad P^{'} \xmapsto{\gamma} S}{P \xmapsto{\gamma} S} \quad \textbf{Res}\ \frac{S \xmapsto{\gamma} S^{'} \qquad y \notin n(\gamma)}{(\nu y)S \xmapsto{\gamma} (\nu y)S^{'}}$$

$$\textbf{Com1}\ \frac{P \xmapsto{\overline{x}y} P^{'} \qquad Q \xmapsto{xy} Q^{'}}{P|Q \xmapsto{\tau} P^{'}|Q^{'}}$$

$$\textbf{Com2LOut}\ \frac{L_1 \xmapsto{\overline{x}y} L_1^{'} \qquad L_2 \xmapsto{xy} S}{L_1|L_2 \xmapsto{\epsilon} L_1^{'}|S} \quad \textbf{Com2ROut}\ \frac{L_1 \xmapsto{xy} S \qquad L_2 \xmapsto{\overline{x}y} L_2^{'}}{L_1|L_2 \xmapsto{\epsilon} S|L_2^{'}}$$

$$\textbf{Com2LInp}\ \frac{L_1 \xmapsto{\overline{x}y} S \qquad L_2 \xmapsto{xy} L_2^{'}}{L_1|L_2 \xmapsto{\epsilon} S|L_2^{'}} \quad \textbf{Com2RInp}\ \frac{L_1 \xmapsto{xy} L_1^{'} \qquad L_2 \xmapsto{\overline{x}y} S}{L_1|L_2 \xmapsto{\epsilon} L_1^{'}|S}$$

$$\textbf{Com3LOut}\ \frac{Q \xmapsto{\overline{x}y} S \qquad P \xmapsto{xy} L}{Q|P \xmapsto{\epsilon} S|L} \quad \textbf{Com3ROut}\ \frac{P \xmapsto{xy} L \qquad Q \xmapsto{\overline{x}y} S}{P|Q \xmapsto{\epsilon} L|S}$$

$$\textbf{Com3LInp}\ \frac{Q \xmapsto{xy} S \qquad P \xmapsto{\overline{x}y} L}{Q|P \xmapsto{\epsilon} S|L} \quad \textbf{Com3RInp}\ \frac{P \xmapsto{\overline{x}y} L \qquad Q \xmapsto{xy} S}{P|Q \xmapsto{\epsilon} L|S}$$

$$\textbf{Com4L}\ \frac{L_1 \xmapsto{\overline{x}y} P \qquad L_2 \xmapsto{xy} Q}{L_1|L_2 \xmapsto{\tau} P|Q} \quad \textbf{Com4R}\ \frac{L_1 \xmapsto{xy} P \qquad L_2 \xmapsto{\overline{x}y} Q}{L_1|L_2 \xmapsto{\tau} P|Q}$$

Table 5.5: Low multi π early semantic with structural congruence

for the inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

A proof of the conclusion is:

$$\textbf{Sum}\ \frac{P_1 \xmapsto{\gamma_1} L_1}{P_1 + P_2 \xmapsto{\gamma_1} L_1}$$

Cong : this case is similar to the previous.

Res : the last part of the derivation tree looks like this:

$$\textbf{Res}\ \frac{P_1 \xrightarrow{\sigma} Q_1 \qquad z \notin n(\sigma)}{(\nu z)P_1 \xrightarrow{\sigma} (\nu z)Q_1}$$

for the inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} Q_1 \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma$$

We can apply the rule *Res* to each of the previous transitions because

$$z \notin n(\sigma) \text{ implies } z \notin n(\gamma_i) \text{ for each } i$$

and then get a proof of the conclusion:

$$(\nu z)P_1 \xmapsto{\gamma_1} (\nu z)L_1 \xmapsto{\gamma_2} (\nu z)L_2 \cdots (\nu z)L_{k-1} \xmapsto{\gamma_k} (\nu z)L_k \xmapsto{\gamma_{k+1}} (\nu z)Q_1$$

Par : this case is similar to the previous.

ECom : the last part of the derivation tree looks like this:

$$\textbf{ECom}\ \frac{P_1 \xrightarrow{\sigma_1} P_1' \qquad Q_1 \xrightarrow{\sigma_2} Q_1' \qquad ESync(\sigma_1, \sigma_2, \sigma_3)}{P_1|Q_1 \xrightarrow{\sigma_3} P_1'|Q_1'}$$

for inductive hypothesis there exist $L_1, \cdots, L_k$ and $\gamma_1, \cdots, \gamma_{k+1}$ with $k \geq 0$ such that

$$P_1 \xmapsto{\gamma_1} L_1 \xmapsto{\gamma_2} L_2 \cdots L_{k-1} \xmapsto{\gamma_k} L_k \xmapsto{\gamma_{k+1}} P_1' \quad \text{and} \quad \gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \sigma_1$$

and there exist $R_1, \cdots, R_h$ and $\delta_1, \cdots, \delta_{h+1}$ with $h \geq 0$ such that

$$Q_1 \xmapsto{\delta_1} R_1 \xmapsto{\delta_2} R_2 \cdots R_{h-1} \xmapsto{\delta_h} R_h \xmapsto{\delta_{h+1}} Q_1' \quad \text{and} \quad \delta_1 \cdot \ldots \cdot \delta_{h+1} = \sigma_2$$

We proceed by cases on the derivation of $ESync(\sigma_1, \sigma_2, \sigma_3)$. We show just some cases because the others are similar.

S1L Suppose that δ_1 is $\overline{x}y$(the other cases are similar), so the other δs are ϵ or τ. We can have three different cases now each :

$\gamma_1 = xy$: The other γs are ϵ or τ. A proof of the conclusion is:

$$P_1|Q_1 \xmapsto{\epsilon} L_1|R_1 \xmapsto{\epsilon} L_2|R_1 \cdots \xmapsto{\epsilon} P_1'|R_1 \xmapsto{\epsilon} P_1'|R_2 \cdots \xmapsto{\epsilon} P_1'|Q_1'$$

we derive the first transition with rule *Com3ROut*, whether for the other transition we use the rules *Par1L*, *Par1R*, *Par3L* or *Par3R*.

$\gamma_i = xy$: A proof of the conclusion is:

$$P_1|Q_1 \xmapsto{\epsilon} L_1|Q_1 \cdots \xmapsto{\epsilon} L_{i-1}|Q_1 \xmapsto{\tau} L_i|R_1 \xmapsto{\epsilon} L_{i+1}|R_1 \cdots \xmapsto{\epsilon} P_1'|R_1 \xmapsto{\epsilon} P_1'|R_2 \cdots \xmapsto{\epsilon} P_1'|Q_1'$$

we derive the transaction $L_{i-1}|Q_1 \overset{\tau}{\longmapsto} L_i|R_1$ with rule $Com5L$, whether for the other transactions we use some rule for parallel.

$\gamma_{k+1} = xy$ similar.

$S2R$: We suppose that $\delta_1 = xy$ and so other δs are ϵ or τ, the other cases are similar. We can have two different cases now depending on where the first $\overline{x}y$ is:

$\gamma_1 = \overline{x}y$: A proof of the conclusion is:

$$P_1|Q_1 \overset{\tau}{\longmapsto} L_1|R_1 \overset{\gamma_2}{\longmapsto} L_2|R_1 \cdots \overset{\gamma_{k+1}}{\longmapsto} P_1'|R_1 \overset{\delta_2}{\longmapsto} P_1'|R_2 \cdots \overset{\delta_{h+1}}{\longmapsto} P_1'|Q_1'$$

we derive the first transition with rule $Com3L$, whether for the other transactions we use some rule for parallel. Since $\gamma_1 \cdot \ldots \cdot \gamma_{k+1} = \overline{x}y \cdot \sigma$ and $\gamma_1 = \overline{x}y$ then $\tau \cdot \gamma_2 \cdot \ldots \cdot \gamma_{k+1} \cdot \epsilon \cdot \ldots \epsilon \cdot \tau = \sigma$

$\gamma_i = \overline{x}y$: A proof of the conclusion is:

$$P_1|Q_1 \overset{\epsilon}{\longmapsto} L_1|Q_1 \cdots \overset{\epsilon}{\longmapsto} L_{i-1}|Q_1 \overset{\tau}{\longmapsto} L_i|R_1 \overset{\gamma_{i+1}}{\longmapsto} L_{i+1}|R_1 \cdots \overset{\gamma_k}{\longmapsto} P_1'|R_1 \overset{\delta_2}{\longmapsto}$$
$$P_1'|R_2 \cdots \overset{\delta_{h+1}}{\longmapsto} P_1'|Q_1'$$

we derive the transition $L_{i-1}|Q_1 \overset{\tau}{\longmapsto} L_i|Q_1'$ with rule $Com2L$, whether for the other transactions of the premises we use the rule $Par1L$.

$\gamma_{k+1} = \overline{x}y$: cannot happen because σ is not empty.

$S4R$ We have three cases: $|\sigma_1| = |\sigma_2|$, $|\sigma_1| > |\sigma_2|$ or $|\sigma_2| > |\sigma_1|$. In the first case $|\sigma_3|$ must be τ and we can build a chain of transition as in the previous cases. In the second case there is a prefix of σ_1 which synchronize with σ_2 and σ_3 is the rest of σ_1, in this case we can also build a chain of transition as in the previous cases. The third case is symmetric to the second.

□

The converse of lemma 5.2.1 does not hold because the low semantic allow to express interleaving behaviour. But there is the following weaker result:

Proposition 5.2.2. Let $\rightarrow$ be the relation defined in table 5.1, let α be an action and P, Q be processes. If $P \overset{\alpha}{\longmapsto} Q$ then $P \overset{\alpha}{\rightarrow} Q$.

Proof. The proof is an easy induction on the proof tree of $P \overset{\alpha}{\longmapsto} Q$. □

Bibliography

[1] Roberto Gorrieri, Cristian Versari, *Multi π: a calculus for mobile multi-party and transactional communication.*

[2] Robin Milner, Joachim Parrow, David Walker, *A calculus of mobile processes, part II*, 1990.

[3] Roberto Gorrieri, *A fully-abstract semantics for atomicity*, Dipartimento di scienze dell'informazione, Universitá di Bologna.

[4] Joachin Parrow, *An introduction to the π calculus*, Department Teleinformatics, Rotal Institute of Technology, Stockholm.

[5] Davide Sangiorgi, David Walker, *The π-calculus*, Cambridge University Press.

[6] Davide Sangiorgi, *A theory of bisimulation for the π-calculus*, Acta informatica, 33(1):69-97, 1996.

[7] Milner, Robin, *Communicating and mobile systems: the π-calculus*, Cambridge University Press.

[8] MohammedReza Mousavi, Michel A Reniers, *Congruence for structural congruences*, Department of Computer Science, Eindhoven University of Technology.