Processos e Concorrência 2015/16
Bloco de acetatos 5

Alexandre Madeira
(based on Luís S. Barbosa 2014/15 course Slides)
HASLab INESC TEC, DI UMINHO

March 26, 2017

Semantics

$$\frac{}{a.p \xrightarrow{a} p}\ (act)$$

$$\frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'}\ (sum - l) \qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'}\ (sum - r)$$

$$\frac{p \xrightarrow{a} p'}{p \mid q \xrightarrow{a} p' \mid q}\ (par - l) \qquad \frac{q \xrightarrow{a} q'}{p \mid q \xrightarrow{a} p \mid q'}\ (par - r)$$

$$\frac{p \xrightarrow{a} p' \quad q \xrightarrow{\overline{a}} q'}{p \mid q \xrightarrow{\tau} p' \mid q'}\ (react) \qquad \frac{p \xrightarrow{a} p'}{p \setminus \{k\} \xrightarrow{a} p' \setminus \{k\}}\ (res)\ \ (\text{if } a \notin \{k, \overline{k}\})$$

$$\frac{p \xrightarrow{a} p'}{p[f] \xrightarrow{f(a)} p'[f]}\ (rel)\ \ (f \text{ relabelling function})$$

$$\frac{p \xrightarrow{a} p'}{k \xrightarrow{a} p'}\ (con)\ \ k =^{df} p$$

Semantics

These rules define a **LTS**

$$\{\xrightarrow{a} \subseteq \mathbb{P} \times \mathbb{P} | a \in Act\}$$

Relation $\xrightarrow{a}$ is defined **inductively** over process structure entailing a semantic description which is

Structural *i.e.*, each process **shape** (defined by the most external combinator) has a type of transitions

Modular *i.e.*, a process transition is defined from transitions in its sup-processes

Complete *i.e.*, all possible transitions are infered from these rules

Graphical representations

Synchronization diagram

- represent interfaces of processes
- static combinators are an **algebra** of synchronization diagrams

Graphical representations

Synchronization diagram

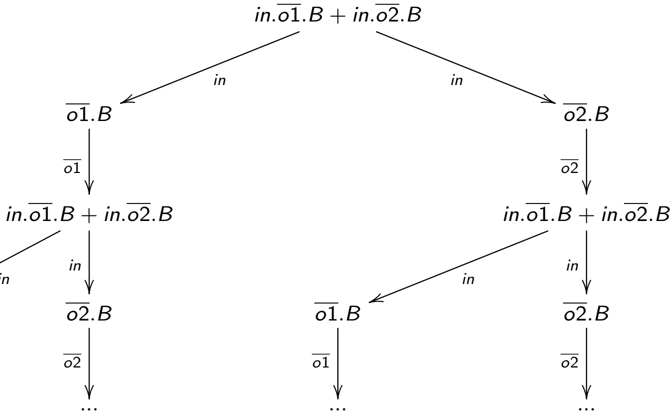
- represent interfaces of processes
- static combinators are an **algebra** of synchronization diagrams

Transition graph

- **derivative**, *n*-**derivative**, **transition tree**
- folds into a transition graph

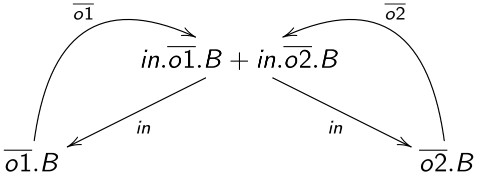
Transition tree

$B =^{df} in.\overline{o1}.B + in.\overline{o2}.B$



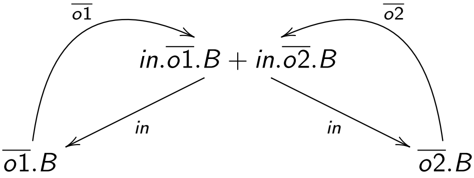
Transition graph

$B =^{df} in.\overline{o1}.B + in.\overline{o2}.B$

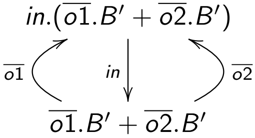


Transition graph

$B =^{df} in.\overline{o1}.B + in.\overline{o2}.B$



compare with $B' =^{df} in.(\overline{o1}.B' + \overline{o2}.B')$



Data parameters

Language $\mathbb{P}$ is extended to $\mathbb{P}_V$ over a data universe V, a set V_e of expressions over V and a evaluation $Val : V_e \rightarrow V$

Example

$$B =^{df} in(x).B'_x$$
$$B'_v =^{df} \overline{out}\langle v\rangle.B$$

Data parameters

Language $\mathbb{P}$ is extended to $\mathbb{P}_V$ over a data universe V, a set V_e of expressions over V and a evaluation $Val : V_e \rightarrow V$

Example

$$B =^{df} in(x).B'_x$$
$$B'_v =^{df} \overline{out}\langle v\rangle.B$$

- Two prefix forms: $a(x).E$ and $\overline{a}\langle e\rangle.E$ (**actions** as **ports**)
- Data parameters: $A_S(x_1, ..., x_n) =^{df} E_A$, with $S \in V$ and each $x_i \in L$
- Conditional combinator: if b then P, if b then P_1 else P_2

Clearly

$$\text{if } b \text{ then } P_1 \text{ else } P_2 \;=^{abv}\; (\text{if } b \text{ then } P_1) + (\text{if } \neg b \text{ then } P_2)$$

Data parameters

Additional semantic rules

$$\frac{}{a(x).E \xrightarrow{a(v)} \{v/x\}E}\ (\mathit{prefix}_i) \qquad \text{for } v \in V$$

$$\frac{}{\overline{a}\langle e\rangle.E \xrightarrow{\overline{a}\langle v\rangle} E}\ (\mathit{prefix}_o) \qquad \text{for } \mathit{Val}(e) = v$$

$$\frac{E_1 \xrightarrow{a} E'}{\text{if } b \text{ then } E_1 \text{ else } E_2 \xrightarrow{a} E'}\ (\mathit{if}_1) \qquad \text{for } \mathit{Val}(b) = \mathbf{tt}$$

$$\frac{E_2 \xrightarrow{a} E'}{\text{if } b \text{ then } E_1 \text{ else } E_2 \xrightarrow{a} E'}\ (\mathit{if}_2) \qquad \text{for } \mathit{Val}(b) = \mathbf{ff}$$

Examples

- $B =^{df} in(x).\overline{out}\langle x\rangle.B$

Examples

- $B =^{df} in(x).\overline{out}\langle x\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle x\rangle.\overline{out}\langle y\rangle.B$

Examples

- $B =^{df} in(x).\overline{out}\langle x\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle x\rangle.\overline{out}\langle y\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle y\rangle.\overline{out}\langle x\rangle.B$

Examples

- $B =^{df} in(x).\overline{out}\langle x\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle x\rangle.\overline{out}\langle y\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle y\rangle.\overline{out}\langle x\rangle.B$
- $B =^{df} in(x).in(y).(\overline{out}\langle y\rangle.B + \overline{out}\langle x\rangle.B)$

Examples

- $B =^{df} in(x).\overline{out}\langle x\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle x\rangle.\overline{out}\langle y\rangle.B$
- $B =^{df} in(x).in(y).\overline{out}\langle y\rangle.\overline{out}\langle x\rangle.B$
- $B =^{df} in(x).in(y).(\overline{out}\langle y\rangle.B + \overline{out}\langle x\rangle.B)$
- $B =^{df} in(x).\overline{out}\langle 2 \times x\rangle.B$
- $B =^{df} in(x).(\textbf{ if } x > 3 \textbf{ then } \overline{out}\langle x\rangle).B$

Back to $\mathbb{P}$

Encoding in the basic language: $\mathcal{T}(\ ) : \mathbb{P}_V \to \mathbb{P}$

$$\begin{aligned}
\mathcal{T}(a(x).E) &= \sum_{v \in V} a_v.\mathcal{T}(\{v/x\}E) \\
\mathcal{T}(\overline{a}\langle e\rangle.E) &= \overline{a}_e.\mathcal{T}(E) \\
\mathcal{T}(\sum_{i \in I} E_i) &= \sum_{i \in I} \mathcal{T}(E_i) \\
\mathcal{T}(E \mid F) &= \mathcal{T}(E) \mid \mathcal{T}(F) \\
\mathcal{T}(E \setminus K) &= \mathcal{T}(E) \setminus \{a_v | a \in K, v \in V\}
\end{aligned}$$

and

$$\mathcal{T}(\text{if } b \text{ then } E) = \begin{cases} \mathcal{T}(E) & \text{if } Val(b) = \mathbf{tt} \\ \mathbf{0} & \text{if } Val(b) = \mathbf{ff} \end{cases}$$

Exercise

Draw the transition diagram of the process *Pred*:

$$Pred =^{df} in(x).Pred'(x)$$

$$Pred'(x) =^{df} \textbf{if } x = 0 \textbf{ then } \overline{out}\langle 0\rangle.Pred \textbf{ else } \overline{out}\langle x - 1\rangle.$$

Semantics

Two-level semantics

- **behavioural** given by transition rules which express how system's components interact (as seen in the last classes)

Semantics

Two-level semantics

- **behavioural** given by transition rules which express how system's components interact (as seen in the last classes)
- **arquitectural**, expresses a notion of **similar assembly configurations** and is expressed through a structural congruence relation;

Structural congruence

$\equiv$ over $\mathbb{P}$ is given by the closure of the following conditions:

- for all $A(\vec{x}) =^{df} E_A$, $A(\vec{y}) \equiv \{\vec{y}/\vec{x}\}\, E_A$,
  (*i.e.*, **folding/unfolding** preserve $\equiv$)
- α-conversion (*i.e.*, replacement of bounded variables).
- both $|$ and $+$ originate, with $\mathbf{0}$, abelian monoids
- forall $a \notin \mathrm{fn}(P)$ $(P \mid Q) \setminus \{a\} \equiv P \mid Q \setminus \{a\}$
- $\mathbf{0} \setminus \{a\} \equiv \mathbf{0}$

Compatibility

Lemma

Structural congruence preserves transitions:

if $p \xrightarrow{a} p'$ and $p \equiv q$ there exists a process q' such that $q \xrightarrow{a} q'$ and $p' \equiv q'$.

Processes are 'prototypical' transition systems

... hence all definitions apply:

$E \sim F$

- Processes E, F are bisimilar if there exist a bisimulation S st $\{\langle E, F\rangle\} \in S$.
- A binary relation S in $\mathbb{P}$ is a (strict) bisimulation iff, whenever $(E, F) \in S$ and $a \in Act$,

$$
\begin{array}{llll}
\text{i)} & E \xrightarrow{a} E' & \Rightarrow & F \xrightarrow{a} F' \text{ and } (E', F') \in S \\
\text{ii)} & F \xrightarrow{a} F' & \Rightarrow & E \xrightarrow{a} E' \text{ and } (E', F') \in S
\end{array}
$$

Alternative characterization of bisimilarity

Recalling bisimilarity definition:

$$\sim = \bigcup\{S \subseteq \mathbb{P} \times \mathbb{P} \mid S \quad \text{is a (strict) bisimulation}\}$$

Alternative characterization of bisimilarity

Recalling bisimilarity definition:

$$\sim = \bigcup \{ S \subseteq \mathbb{P} \times \mathbb{P} \mid S \quad \text{is a (strict) bisimulation} \}$$

Usefull Lemma:

$E \sim F$ iff

$$\begin{array}{rlcl} \text{i)} & E \xrightarrow{a} E' & \Rightarrow & F \xrightarrow{a} F' \text{ and } E' \sim F' \\ \text{ii)} & F \xrightarrow{a} F' & \Rightarrow & E \xrightarrow{a} E' \text{ and } E' \sim F' \end{array}$$

Processes are ‘prototypical’ transition systems

Example: $S \sim M$

$$
\begin{aligned}
T &=^{df} i.\overline{k}.T \\
R &=^{df} k.j.R \\
S &=^{df} (T \mid R) \setminus \{k\}
\end{aligned}
$$

$$
\begin{aligned}
M &=^{df} i.\tau.N \\
N &=^{df} j.i.\tau.N + i.j.\tau.N
\end{aligned}
$$

Processes are 'prototypical' transition systems

Example: $S \sim M$

$$T =^{df} i.\overline{k}.T$$
$$R =^{df} k.j.R$$
$$S =^{df} (T \mid R) \setminus \{k\}$$

$$M =^{df} i.\tau.N$$
$$N =^{df} j.i.\tau.N + i.j.\tau.N$$

through bisimulation

$$R = \{\langle S, M\rangle, \langle(\overline{k}.T \mid R) \setminus \{k\}, \tau.N\rangle, \langle(T \mid j.R) \setminus \{k\}, N\rangle, \langle(\overline{k}.T \mid j.R) \setminus \{k\}, j.\tau.N\rangle\}$$

Example: Semaphores

A semaphore

$$Sem \;=^{df}\; get.put.Sem$$

Example: Semaphores

A semaphore

$$Sem \;=^{df}\; get.put.Sem$$

n-semaphores

$$\begin{aligned}
Sem_n &=^{df} Sem_{n,0} \\
Sem_{n,0} &=^{df} get.Sem_{n,1} \\
Sem_{n,i} &=^{df} get.Sem_{n,i+1} + put.Sem_{n,i-1} \\
&\quad\;\; (\text{for } 0 < i < n) \\
Sem_{n,n} &=^{df} put.Sem_{n,n-1}
\end{aligned}$$

Example: Semaphores

A semaphore

$$Sem \;=^{df}\; get.put.Sem$$

n-semaphores

$$\begin{aligned}
Sem_n &=^{df} Sem_{n,0} \\
Sem_{n,0} &=^{df} get.Sem_{n,1} \\
Sem_{n,i} &=^{df} get.Sem_{n,i+1} + put.Sem_{n,i-1} \\
&\quad (\text{for } 0 < i < n) \\
Sem_{n,n} &=^{df} put.Sem_{n,n-1}
\end{aligned}$$

Sem_n can also be implemented by the parallel composition of n Sem processes:

$$Sem^n \;=^{df}\; Sem \mid Sem \mid \ldots \mid Sem$$

Example: Semaphores

Is $Sem_n \sim Sem^n$?

For $n = 2$:

$$\{\langle Sem_{2,0}, Sem \mid Sem\rangle, \langle Sem_{2,1}, Sem \mid put.Sem\rangle,$$
$$\langle Sem_{2,1}, put.Sem \mid Sem\rangle\langle Sem_{2,2}, put.Sem \mid put.Sem\rangle\}$$

is a bisimulation.

Example: Semaphores

Is $Sem_n \sim Sem^n$?

For $n = 2$:

$$\{\langle Sem_{2,0}, Sem \mid Sem\rangle, \langle Sem_{2,1}, Sem \mid put.Sem\rangle,$$
$$\langle Sem_{2,1}, put.Sem \mid Sem\rangle \langle Sem_{2,2}, put.Sem \mid put.Sem\rangle\}$$

is a bisimulation.

- but can we get rid of structurally congruent pairs?

Bisimulation up to $\equiv$

Definition

A binary relation S in $\mathbb{P}$ is a (strict) bisimulation up to $\equiv$ iff, whenever $(E, F) \in S$ and $a \in Act$,

$$\text{i)} \quad E \xrightarrow{a} E' \quad \Rightarrow \quad F \xrightarrow{a} F' \text{ and } (E', F') \in \equiv \cdot S \cdot \equiv$$

$$\text{ii)} \quad F \xrightarrow{a} F' \quad \Rightarrow \quad E \xrightarrow{a} E' \text{ and } (E', F') \in \equiv \cdot S \cdot \equiv$$

Bisimulation up to $\equiv$

Definition

A binary relation S in $\mathbb{P}$ is a (strict) bisimulation up to $\equiv$ iff, whenever $(E, F) \in S$ and $a \in Act$,

$$\text{i)} \quad E \xrightarrow{a} E' \;\Rightarrow\; F \xrightarrow{a} F' \text{ and } (E', F') \in \equiv \cdot S \cdot \equiv$$

$$\text{ii)} \quad F \xrightarrow{a} F' \;\Rightarrow\; E \xrightarrow{a} E' \text{ and } (E', F') \in \equiv \cdot S \cdot \equiv$$

Lemma

If S is a (strict) bisimulation up to $\equiv$, then $S \subseteq \sim$

A $\sim$-calculus

Lemma

$$E \equiv F \;\Rightarrow\; E \sim F$$

$\sim$ is a congruence

congruence is the name of **modularity** in Mathematics

- **process combinators** preserve $\sim$

$\sim$ is a congruence

congruence is the name of **modularity** in Mathematics

- **process combinators** preserve $\sim$

Lemma

Assume $E \sim F$. Then,

$$a.E \sim a.F$$
$$E + P \sim F + P$$
$$E \mid P \sim F \mid P$$
$$E \setminus K \sim F \setminus K$$
$$E[f] \sim F[f]$$

$\sim$ is a congruence

congruence is the name of **modularity** in Mathematics

- **process combinators** preserve $\sim$

Lemma

Assume $E \sim F$. Then,

$$a.E \sim a.F$$
$$E + P \sim F + P$$
$$E \mid P \sim F \mid P$$
$$E \setminus K \sim F \setminus K$$
$$E[f] \sim F[f]$$

- **recursive definition** preserves $\sim$

$\sim$ is a congruence

- First $\sim$ is extended to processes with variables:

$$E \sim F \;\equiv\; \forall_{\tilde{P}} .\; E[\tilde{P}/\tilde{X}] \sim F[\tilde{P}/\tilde{X}]$$

- Then prove:

Lemma

i) $\tilde{P} =^{df} \tilde{E} \;\Rightarrow\; \tilde{P} \sim \tilde{E}$
where $\tilde{E}$ is a family of process expressions and $\tilde{P}$ a family of process **identifiers**.

ii) Let $\tilde{E} \sim \tilde{F}$, where $\tilde{E}$ and $\tilde{F}$ are families of recursive process expressions over a family of process **variables** $\tilde{X}$, and define:

$$\tilde{A} =^{df} \tilde{E}[\tilde{A}/\tilde{X}] \text{ and } \tilde{B} =^{df} \tilde{F}[\tilde{B}/\tilde{X}]$$

Then

$$\tilde{A} \sim \tilde{B}$$

The expansion theorem

Every process is equivalent to the sum of its derivatives

$$E \sim \sum \{a.E' \mid E \xrightarrow{a} E'\}$$

The expansion theorem

The usual definition (based on the **concurrent canonical form**):

$$E \;\sim\; \sum \{\; f_i(a).(E_1[f_1] \mid \ldots \mid E_i'[f_i] \mid \ldots \mid E_n[f_n]) \setminus K \;\;\mid\;\; E_i \xrightarrow{a} E_i' \;\text{ and }\; f_i(a) \notin K \cup \overline{K} \;\}$$
$$+$$
$$\sum \{\; \tau.(E_1[f_1] \mid \ldots \mid E_i'[f_i] \mid \ldots \mid E_j'[f_j] \mid \ldots \mid E_n[f_n]) \setminus K \;\;\mid\;\; E_i \xrightarrow{a} E_i' \;\text{ and }\; E_j \xrightarrow{b} E_j' \;\text{ and }\; f_i(a) = \overline{f_j(b)} \;\}$$

for $E =^{df} (E_1[f_1] \mid \ldots \mid E_n[f_n]) \setminus K$, with $n \geq 1$

The expansion theorem

Corollary (for $n = 1$ and $f_1 = id$)

$$(E + F) \setminus K \sim E \setminus K + F \setminus K$$

$$(a.E) \setminus K \sim \begin{cases} \mathbf{0} & \text{if } a \in (K \cup \overline{K}) \\ a.(E \setminus K) & \text{otherwise} \end{cases}$$

Revisit the example and show $S \sim M$ using the expansion theorem

$$
\begin{aligned}
T &=^{df} i.\overline{k}.T \\
R &=^{df} k.j.R \\
S &=^{df} (T \mid R) \setminus \{k\}
\end{aligned}
$$

$$
\begin{aligned}
M &=^{df} i.\tau.N \\
N &=^{df} j.i.\tau.N + i.j.\tau.N
\end{aligned}
$$

Example

$S \sim M$

Example

$S \sim M$

$$
\begin{aligned}
S &\sim (T \mid R) \setminus \{k\} \\
&\sim i.(\overline{k}.T \mid R) \setminus \{k\} \\
&\sim i.\tau.(T \mid j.R) \setminus \{k\} \\
&\sim i.\tau.(i.\,(\overline{k}.T \mid j.R) \setminus \{k\} + j.(T \mid R) \setminus \{k\}) \\
&\sim i.\tau.(i.j.\,(\overline{k}.T \mid R) \setminus \{k\} + j.i.(\overline{k}.T \mid R) \setminus \{k\}) \\
&\sim i.\tau.(i.j.\tau.\,(T \mid j.R) \setminus \{k\} + j.i.\tau.(T \mid j.R) \setminus \{k\})
\end{aligned}
$$

Example

$S \sim M$

$$
\begin{aligned}
S &\sim (T \mid R) \setminus \{k\} \\
&\sim i.(\overline{k}.T \mid R) \setminus \{k\} \\
&\sim i.\tau.(T \mid j.R) \setminus \{k\} \\
&\sim i.\tau.(i.\,(\overline{k}.T \mid j.R) \setminus \{k\} + j.(T \mid R) \setminus \{k\}) \\
&\sim i.\tau.(i.j.\,(\overline{k}.T \mid R) \setminus \{k\} + j.i.(\overline{k}.T \mid R) \setminus \{k\}) \\
&\sim i.\tau.(i.j.\tau.\,(T \mid j.R) \setminus \{k\} + j.i.\tau.(T \mid j.R) \setminus \{k\})
\end{aligned}
$$

Let $N' = (T \mid j.R) \setminus \{k\}$.
This expands into $N' \sim i.j.\tau.\,(T \mid j.R) \setminus \{k\} + j.i.\tau.(T \mid j.R) \setminus \{k\}$,
Therefore $N' \sim N$ and $S \sim i.\tau.N \sim M$

- requires result on unique solutions for recursive process equations

Exercise

Using the expansion theorem, reduce P and Q into its concurrent normal form

$$P_1 =^{df} a.P_1' + b.P_2''$$
$$P_2 =^{df} \overline{a}.P_2' + c.P_2''$$
$$P_3 =^{df} \overline{a}.P_3' + \overline{c}.P_3''$$
$$P =^{df} (P_1 \mid P_2) \setminus \{a\}$$
$$Q =^{df} (P_1 \mid P_2 \mid P_3) \setminus \{a, b\}$$

Observable transitions

$$\stackrel{a}{\Rightarrow} \subseteq \mathbb{P} \times \mathbb{P}$$

- $L \cup \{\epsilon\}$
- A $\stackrel{\epsilon}{\Rightarrow}$-transition corresponds to zero or more **non observable** transitions
- inference rules for $\stackrel{a}{\Rightarrow}$:

$$\frac{}{E \stackrel{\epsilon}{\Rightarrow} E}\,(O_1)$$

$$\frac{E \stackrel{\tau}{\to} E' \quad E' \stackrel{\epsilon}{\Rightarrow} F}{E \stackrel{\epsilon}{\Rightarrow} F}\,(O_2)$$

$$\frac{E \stackrel{\epsilon}{\Rightarrow} E' \quad E' \stackrel{a}{\to} F' \quad F' \stackrel{\epsilon}{\Rightarrow} F}{E \stackrel{a}{\Rightarrow} F}\,(O_3) \quad \text{for } a \in L$$

Example

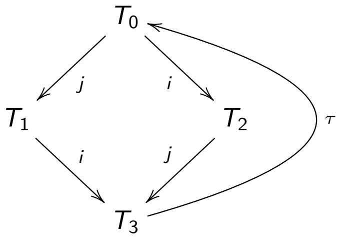
$$T_0 =^{df} j.T_1 + i.T_2$$
$$T_1 =^{df} i.T_3$$
$$T_2 =^{df} j.T_3$$
$$T_3 =^{df} \tau.T_0$$

and

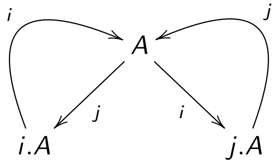
$$A =^{df} i.j.A + j.i.A$$

Example

From their graphs,



and



we conclude that $T_0 \not\approx A$ (why?).

Observational equivalence

$E \approx F$

- A binary relation S in $\mathbb{P}$ is a weak bisimulation iff, whenever $(E, F) \in S$ and $a \in L \cup \{\epsilon\}$,

$$\begin{array}{rl} \text{i)} & E \stackrel{a}{\Rightarrow} E' \quad \Rightarrow \quad F \stackrel{a}{\Rightarrow} F' \text{ and } (E', F') \in S \\ \text{ii)} & F \stackrel{a}{\Rightarrow} F' \quad \Rightarrow \quad E \stackrel{a}{\Rightarrow} E' \text{ and } (E', F') \in S \end{array}$$

Observational equivalence

$E \approx F$

- A binary relation S in $\mathbb{P}$ is a weak bisimulation iff, whenever $(E, F) \in S$ and $a \in L \cup \{\epsilon\}$,

$$\begin{array}{rl} \text{i)} & E \stackrel{a}{\Rightarrow} E' \quad \Rightarrow \quad F \stackrel{a}{\Rightarrow} F' \;\text{ and }\; (E', F') \in S \\ \text{ii)} & F \stackrel{a}{\Rightarrow} F' \quad \Rightarrow \quad E \stackrel{a}{\Rightarrow} E' \;\text{ and }\; (E', F') \in S \end{array}$$

- Processes E, F are observationally equivalent if there exists a weak bisimulation S st $\{\langle E, F\rangle\} \in S$

Observational equivalence

$E \approx F$

- A binary relation S in $\mathbb{P}$ is a weak bisimulation iff, whenever $(E, F) \in S$ and $a \in L \cup \{\epsilon\}$,

$$\begin{array}{rl} \text{i)} & E \overset{a}{\Rightarrow} E' \quad \Rightarrow \quad F \overset{a}{\Rightarrow} F' \ \text{ and } \ (E', F') \in S \\ \text{ii)} & F \overset{a}{\Rightarrow} F' \quad \Rightarrow \quad E \overset{a}{\Rightarrow} E' \ \text{ and } \ (E', F') \in S \end{array}$$

- Processes E, F are observationally equivalent if there exists a weak bisimulation S st $\{\langle E, F \rangle\} \in S$
I.e.,

$$\approx \; = \bigcup \{S \subseteq \mathbb{P} \times \mathbb{P} \mid S \ \text{ is a } \textbf{weak} \text{ bisimulation}\}$$

Properties

- **as expected**: $\approx$ is an equivalence relation

Properties

- **as expected**: $\approx$ is an equivalence relation
- **basic property**: for any $E \in \mathbb{P}$,

$$E \approx \tau.E$$

(**proof idea**: $\mathrm{id}_{\mathbb{P}} \cup \{(E, \tau.E) \mid E \in \mathbb{P}\}$ is a weak bisimulation

Properties

- **as expected**: $\approx$ is an equivalence relation
- **basic property**: for any $E \in \mathbb{P}$,

$$E \;\approx\; \tau.E$$

(**proof idea**: $\mathrm{id}_{\mathbb{P}} \cup \{(E, \tau.E) \mid E \in \mathbb{P}\}$ is a weak bisimulation

- **weak** vs. **strict**:

$$\sim \subseteq \approx$$

Is $\approx$ a congruence?

Lemma

Let $E \approx F$. Then, for any $P \in \mathbb{P}$ and $K \subseteq L$,

$$a.E \approx a.F$$

$$E \mid P \approx F \mid P$$

$$E \setminus K \approx F \setminus K$$

$$E[f] \approx F[f]$$

Is $\approx$ a congruence?

Lemma
Let $E \approx F$. Then, for any $P \in \mathbb{P}$ and $K \subseteq L$,

$$a.E \approx a.F$$
$$E \mid P \approx F \mid P$$
$$E \setminus K \approx F \setminus K$$
$$E[f] \approx F[f]$$

but

$$E + P \approx F + P$$

does not hold, in general.

Is $\approx$ a congruence?

Example (initial τ restricts options menu')

$$i.\mathbf{0} \approx \tau.i.\mathbf{0}$$

Is $\approx$ a congruence?

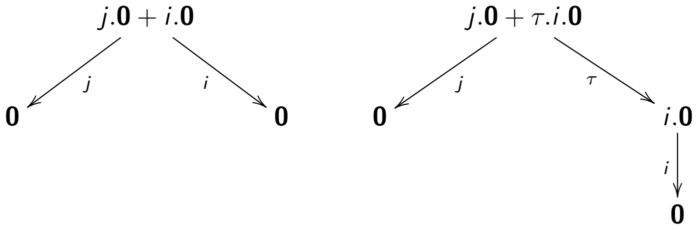
Example (initial τ restricts options menu')

$$i.\mathbf{0} \;\approx\; \tau.i.\mathbf{0}$$

However

$$j.\mathbf{0} + i.\mathbf{0} \not\approx j.\mathbf{0} + \tau.i.\mathbf{0}$$

Actually,



Forcing a congruence: $E = F$

Solution: force any initial τ to be matched by another τ

Process equality

Two processes E and F are equal (or **observationally congruent**) iff

i) $E \approx F$

ii) $E \xrightarrow{\tau} E' \quad \Rightarrow \quad F \xrightarrow{\tau} X \overset{\epsilon}{\Rightarrow} F'$ and $E' \approx F'$

iii) $F \xrightarrow{\tau} F' \quad \Rightarrow \quad E \xrightarrow{\tau} X \overset{\epsilon}{\Rightarrow} E'$ and $E' \approx F'$

Forcing a congruence: $E = F$

Solution: force any initial τ to be matched by another τ

Process equality

Two processes E and F are equal (or **observationally congruent**) iff

i) $E \approx F$

ii) $E \xrightarrow{\tau} E' \;\Rightarrow\; F \xrightarrow{\tau} X \stackrel{\epsilon}{\Rightarrow} F'$ and $E' \approx F'$

iii) $F \xrightarrow{\tau} F' \;\Rightarrow\; E \xrightarrow{\tau} X \stackrel{\epsilon}{\Rightarrow} E'$ and $E' \approx F'$

- note that $E \neq \tau.E$, but $\tau.E = \tau.\tau.E$

Forcing a congruence: $E = F$

$=$ can be regarded as a restriction of $\approx$ to all pairs of processes which preserve it in **additive** contexts

Lemma

Let E and F be processes st the union of their sorts is distinct of L. Then,

$$E = F \quad \equiv \quad \forall_{G \in \mathbb{P}} \,.\, (E + G \;\approx\; F + G)$$

Properties of $=$

Lemma

$$E \approx F \;\equiv\; (E = F) \vee (E = \tau.F) \vee (\tau.E = F)$$

Properties of $=$

Lemma

$$\sim \; \subseteq \; = \; \subseteq \; \approx$$

So,

$$\boxed{\text{the whole} \sim \text{theory remains valid}}$$

Additionally,

Lemma (additional laws)

$$\begin{aligned} a.\tau.E &= a.E \\ E + \tau.E &= \tau.E \\ a.(E + \tau.F) &= a.(E + \tau.F) + a.F \end{aligned}$$

Conditions on variables

guarded :
X occurs in a sub-expression of type $a.E'$ for $a \in Act - \{\tau\}$

weakly guarded :
X occurs in a sub-expression of type $a.E'$ for $a \in Act$

in both cases assures that, until a guard is reached, behaviour does not depends on the process that instantiates the variable

Conditions on variables

guarded :
: X occurs in a sub-expression of type $a.E'$ for $a \in Act - \{\tau\}$

weakly guarded :
: X occurs in a sub-expression of type $a.E'$ for $a \in Act$

in both cases assures that, until a guard is reached, behaviour does not depends on the process that instantiates the variable

example: X is weakly guarded in both $\tau.X$ and $\tau.\mathbf{0} + a.X + b.a.X$ but guarded only in the second

Conditions on variables

sequential :

X is sequential in E if every **strict** sub-expression in which X occurs is either $a.E'$, for $a \in Act$, or $\Sigma\tilde{E}$.

avoids X to become guarded by a τ as a result of an interaction

Conditions on variables

sequential :

X is sequential in E if every **strict** sub-expression in which X occurs is either $a.E'$, for $a \in Act$, or $\Sigma \tilde{E}$.

avoids X to become guarded by a τ as a result of an interaction

example: X is not sequential in $X = (\bar{a}.X \mid a.\mathbf{0}) \setminus \{a\}$

Solving equations

Have equations over $(\mathbb{P}, \sim)$ or $(\mathbb{P}, =)$ (unique) solutions?

Solving equations

Have equations over $(\mathbb{P}, \sim)$ or $(\mathbb{P}, =)$ (unique) solutions?

Lemma

Recursive equations $\tilde{X} = \tilde{E}(\tilde{X})$ or $\tilde{X} \sim \tilde{E}(\tilde{X})$, over $\mathbb{P}$, have unique solutions (up to $=$ or $\sim$, respectively). Formally,

i) Let $\tilde{E} = \{E_i \mid i \in I\}$ be a family of expressions with a maximum of I free variables ($\{X_i \mid i \in I\}$) such that any variable free in E_i is weakly guarded. Then

$$\tilde{P} \sim \{\tilde{P}/\tilde{X}\}\tilde{E} \ \wedge \ \tilde{Q} \sim \{\tilde{Q}/\tilde{X}\}\tilde{E} \quad \Rightarrow \quad \tilde{P} \sim \tilde{Q}$$

Solving equations

Have equations over $(\mathbb{P}, \sim)$ or $(\mathbb{P}, =)$ (unique) solutions?

Lemma

Recursive equations $\tilde{X} = \tilde{E}(\tilde{X})$ or $\tilde{X} \sim \tilde{E}(\tilde{X})$, over $\mathbb{P}$, have unique solutions (up to $=$ or $\sim$, respectively). Formally,

i) Let $\tilde{E} = \{E_i \mid i \in I\}$ be a family of expressions with a maximum of I free variables ($\{X_i \mid i \in I\}$) such that any variable free in E_i is weakly guarded. Then

$$\tilde{P} \sim \{\tilde{P}/\tilde{X}\}\tilde{E} \ \wedge \ \tilde{Q} \sim \{\tilde{Q}/\tilde{X}\}\tilde{E} \ \Rightarrow \ \tilde{P} \sim \tilde{Q}$$

ii) Let $\tilde{E} = \{E_i \mid i \in I\}$ be a family of expressions with a maximum of I free variables ($\{X_i \mid i \in I\}$) such that any variable free in E_i is guarded and sequential. Then

$$\tilde{P} = \{\tilde{P}/\tilde{X}\}\tilde{E} \ \wedge \ \tilde{Q} = \{\tilde{Q}/\tilde{X}\}\tilde{E} \ \Rightarrow \ \tilde{P} = \tilde{Q}$$

Example (1)

Consider

$$
\begin{aligned}
Sem &=^{df} get.put.Sem \\
P_1 &=^{df} \overline{get}.c_1.\overline{put}.P_1 \\
P_2 &=^{df} \overline{get}.c_2.\overline{put}.P_2 \\
S &=^{df} (Sem \mid P_1 \mid P_2) \setminus \{get, put\}
\end{aligned}
$$

and

$$S' =^{df} \tau.c_1.S' + \tau.c_2.S'$$

in order to prove $S = S'$:

Example (1)

Consider

$$
\begin{aligned}
Sem &=^{df} get.put.Sem \\
P_1 &=^{df} \overline{get}.c_1.\overline{put}.P_1 \\
P_2 &=^{df} \overline{get}.c_2.\overline{put}.P_2 \\
S &=^{df} (Sem \mid P_1 \mid P_2) \setminus \{get, put\}
\end{aligned}
$$

and

$$S' =^{df} \tau.c_1.S' + \tau.c_2.S'$$

in order to prove $S = S'$: it is enough to show that both are **solutions** of

$$X \;=\; \tau.c_1.X + \tau.c_2.X$$

Example (1)

Then:

$$
\begin{aligned}
S = {} & \tau.\,(c_1.\overline{put}.P_1 \mid P_2 \mid put.Sem) \setminus K + \tau.(P_1 \mid c_2.\overline{put}.P_2 \mid put.Sem) \setminus K \\
= {} & \tau.c_1.\,(\overline{put}.P_1 \mid P_2 \mid put.Sem) \setminus K + \tau.c_2.(P_1 \mid \overline{put}.P_2 \mid put.Sem) \setminus K \\
= {} & \tau.c_1.\tau.\,(P_1 \mid P_2 \mid Sem) \setminus K + \tau.c_2.\tau.(P_1 \mid P_2 \mid Sem) \setminus K \\
= {} & \tau.c_1.\tau.S + \tau.c_2.\tau.S \\
= {} & \tau.c_1.S + \tau.c_2.S \\
= {} & \{S/X\}E
\end{aligned}
$$

Example (1)

Then:

$$
\begin{aligned}
S = \; & \tau.\,(c_1.\overline{put}.P_1 \mid P_2 \mid put.Sem) \setminus K + \tau.(P_1 \mid c_2.\overline{put}.P_2 \mid put.Sem) \setminus K \\
= \; & \tau.c_1.\,(\overline{put}.P_1 \mid P_2 \mid put.Sem) \setminus K + \tau.c_2.(P_1 \mid \overline{put}.P_2 \mid put.Sem) \setminus K \\
= \; & \tau.c_1.\tau.\,(P_1 \mid P_2 \mid Sem) \setminus K + \tau.c_2.\tau.(P_1 \mid P_2 \mid Sem) \setminus K \\
= \; & \tau.c_1.\tau.S + \tau.c_2.\tau.S \\
= \; & \tau.c_1.S + \tau.c_2.S \\
= \; & \{S/X\}E
\end{aligned}
$$

for S' is immediate

Example (2)

Consider,

$$B =^{df} in.B_1$$
$$B_1 =^{df} in.B_2 + \overline{out}.B$$
$$B_2 =^{df} \overline{out}.B_1$$

$$B' =^{df} (C_1 \mid C_2) \setminus m$$
$$C_1 =^{df} in.\overline{m}.C_1$$
$$C_2 =^{df} m.\overline{out}.C_2$$

Example (2)

Consider,

$$B =^{df} in.B_1 \qquad B' =^{df} (C_1 \mid C_2) \setminus m$$

$$B_1 =^{df} in.B_2 + \overline{out}.B \qquad C_1 =^{df} in.\overline{m}.C_1$$

$$B_2 =^{df} \overline{out}.B_1 \qquad C_2 =^{df} m.\overline{out}.C_2$$

B is a solution of

$$X = E(X, Y, Z) = in.Y$$

$$Y = E_1(X, Y, Z) = in.Z + \overline{out}.X$$

$$Z = E_3(X, Y, Z) = \overline{out}.Y$$

through $\sigma = \{B/X, B_1/Y, B_2/Z\}$

Example (2)

To prove B =B'

$$
\begin{aligned}
B' &= (C_1 \mid C_2) \setminus m \\
&= in.(\overline{m}.C_1 \mid C_2) \setminus m \\
&= in.\tau.(C_1 \mid \overline{out}.C_2) \setminus m \\
&= in.(C_1 \mid \overline{out}.C_2) \setminus m
\end{aligned}
$$

Let $S_1 = (C_1 \mid \overline{out}.C_2) \setminus m$ to proceed:

$$
\begin{aligned}
S_1 &= (C_1 \mid \overline{out}.C_2) \setminus m \\
&= in.\,(\overline{m}.C_1 \mid \overline{out}.C_2) \setminus m + \overline{out}.(C_1 \mid C_2) \setminus m \\
&= in.\,(\overline{m}.C_1 \mid \overline{out}.C_2) \setminus m + \overline{out}.B'
\end{aligned}
$$

Example (2)

Finally, let, $S_2 = (\overline{m}.C_1 \mid \overline{out}.C_2) \setminus m$. Then,

$$\begin{aligned} S_2 &= (\overline{m}.C_1 \mid \overline{out}.C_2) \setminus m \\ &= \overline{out}.(\overline{m}.C_1 \mid C_2) \setminus m \\ &= \overline{out}.\tau.(C_1 \mid \overline{out}.C_2) \setminus m \\ &= \overline{out}.\tau.S_1 \\ &= \overline{out}.S_1 \end{aligned}$$

Example (2)

Note the same problem can be solved with a system of 2 equations:

$$
\begin{aligned}
X =&\ E(X, Y) = in.Y \\
Y =&\ E'(X, Y) = in.\overline{out}.Y + \overline{out}.in.Y
\end{aligned}
$$

Clearly, by substitution,

$$
\begin{aligned}
B =&\ in.B_1 \\
B_1 =&\ in.\overline{out}.B_1 + \overline{out}.in.B_1
\end{aligned}
$$

Example (2)

On the other hand, it's already proved that $B' = \ldots = in.S_1$.
so,

$$
\begin{aligned}
S_1 &= (C_1 \mid \overline{out}.C_2) \setminus m \\
&= in.\,(\overline{m}.C_1 \mid \overline{out}.C_2) \setminus m + \overline{out}.B' \\
&= in.\overline{out}.\,(\overline{m}.C_1 \mid C_2) \setminus m + \overline{out}.B' \\
&= in.\overline{out}.\tau.\,(C_1 \mid \overline{out}.C_2) \setminus m + \overline{out}.B' \\
&= in.\overline{out}.\tau.S_1 + \overline{out}.B' \\
&= in.\overline{out}.S_1 + \overline{out}.B' \\
&= in.\overline{out}.S_1 + \overline{out}.in.S_1
\end{aligned}
$$

Hence, $B' = \{B'/X, S_1/Y\}E$ and $S_1 = \{B'/X, S_1/Y\}E'$

Exercises

Suppose two variants of parallel composition have been added to the process language $\mathbb{P}$ and defined through the following rules:

$$\frac{E \xrightarrow{a} E'}{E \otimes F \xrightarrow{a} E' \otimes F}\ (O_1) \qquad \frac{F \xrightarrow{a} F'}{E \otimes F \xrightarrow{a} E \otimes F'}\ (O_2)$$

$$\frac{E \xrightarrow{a} E' \quad \text{and} \quad \overline{a} \notin \mathcal{L}(F)}{E \parallel F \xrightarrow{a} E' \parallel F}\ (P_1) \qquad \frac{F \xrightarrow{a} F' \quad \text{and} \quad \overline{a} \notin \mathcal{L}(E)}{E \parallel F \xrightarrow{a} E \parallel F'}\ (P_2)$$

$$\frac{E \xrightarrow{a} E' \quad F \xrightarrow{\overline{a}} F'}{E \parallel F \xrightarrow{\tau} E' \parallel F'}\ (P_3)$$

1. Explain, in your own words, the meaning of $\otimes$ e $\parallel$.
2. prove or refute:
   - $\otimes$ is associative with respect to $\sim$
   - $\parallel$ is associative with respect to $\sim$

Exercise

Consider the following statements about a binary relation S on $\mathbb{P}$. Discuss whether you may conclude from each of them whether S is (or is not) a weak bisimulation:

1. S is the identity in $\mathbb{P}$.
2. S is a subset of the identity in $\mathbb{P}$.
3. S is a strict bisimulation up to $\equiv$.
4. S is the empty relation.
5. $S = \{(a.E, a.F) \mid E \approx F\}$.
6. $S = \{(a.E, a.F) \mid E \approx F\} \cup \approx$.

Exercise

Suppose processes R and T have transitions $R \xrightarrow{\tau} T$ and $T \xrightarrow{\tau} R$, among others. Show that, under this condition, $R = T$.

Identify, in the list of process pairs below, which of them can be related by $\approx$. And by $=$?

1. $a.\tau.b.\mathbf{0}$ e $a.b.\mathbf{0}$
2. $a.(b.\mathbf{0} + \tau.c.\mathbf{0})$ e $a.(b.\mathbf{0} + c.\mathbf{0})$
3. $a.(b.\mathbf{0} + \tau.c.\mathbf{0})$ e $a.(b.\mathbf{0} + c.\mathbf{0}) + a.c.\mathbf{0}$
4. $a.\mathbf{0} + b.\mathbf{0} + \tau.b.\mathbf{0}$ e $a.\mathbf{0} + \tau.b.\mathbf{0}$
5. $a.\mathbf{0} + b.\mathbf{0} + \tau.b.\mathbf{0}$ e $a.\mathbf{0} + b.\mathbf{0}$
6. $a.(b.\mathbf{0} + (\tau.(c.\mathbf{0} + \tau.d.\mathbf{0})))$ e
   $a.(b.\mathbf{0} + (\tau.(c.\mathbf{0} + \tau.d.\mathbf{0}))) + a.(c.\mathbf{0} + \tau.d.\mathbf{0})$
7. $a.(b.\mathbf{0} + (\tau.(c.\mathbf{0} + \tau.d.\mathbf{0})))$ e
   $a.(b.\mathbf{0} + c.\mathbf{0} + d.\mathbf{0}) + a.(c.\mathbf{0} + d.\mathbf{0}) + a.d.\mathbf{0}$
8. $\tau.(a.b.\mathbf{0} + a.c.\mathbf{0})$ e $\tau.a.b.\mathbf{0} + \tau.a.c.\mathbf{0}$
9. $\tau.(a.\tau.b.\mathbf{0} + a.b.\tau.\mathbf{0})$ e $a.b.\mathbf{0}$
10. $\tau.(\tau.a.\mathbf{0} + \tau.b.\mathbf{0})$ e $\tau.a.\mathbf{0} + \tau.b.\mathbf{0}$
11. $A =^{df} a.\tau.A$ e $B =^{df} a.B$
12. $A =^{df} \tau.A + a.\mathbf{0}$ e $a.\mathbf{0}$
13. $A =^{df} \tau.A$ e $\mathbf{0}$

Consider the following specification of a *pipe*, as supported e.g. in UNIX:

$$U \triangleright V \;=^{abv}\; (U[c/out] \mid V[c/in]) \setminus \{c\}$$

under the assumption that, in both processes, actions $\overline{out}$ e *in* stand for, respectively, the output and input ports.

1. Consider now the following processes only partially defined:

$$\begin{aligned} U_1 &=^{df} \overline{out}.T \\ V_1 &=^{df} in.R \\ U_2 &=^{df} \overline{out}.\overline{out}.\overline{out}.T \\ V_2 &=^{df} in.in.in.R \end{aligned}$$

   Prove, by equational reasoning, or refute the following properties:

   1. $U_1 \triangleright V_1 \;\sim\; T \triangleright R$
   2. $U_2 \triangleright V_2 \;=\; U_1 \triangleright V_1$

2. Show that $\mathbf{0} \triangleright \mathbf{0} \;=\; \mathbf{0}$.