Mobile Computing

The π-calculus - Equational theory

Pascal Zimmer

pzimmer@daimi.au.dk

BRICS

Reminder – Syntax

$$
\begin{array}{rcl}
P & ::= & \mathbf{0} \\
 & | & a(x).P \\
 & | & \bar{a}x.P \\
 & | & P_1 \mid P_2 \\
 & | & (\nu a)P \\
 & | & !P \\
 & | & P + Q
\end{array}
$$

Operational Semantics

$$\frac{}{(M + x(y).P) \mid (N + \bar{x}z.Q) \;\rightarrow\; P\{y \mapsto z\} \mid Q}\,\text{(Com)}$$

$$\frac{P \;\rightarrow\; P'}{P \mid Q \;\rightarrow\; P' \mid Q}\,\text{(Par)} \qquad \frac{P \;\rightarrow\; P'}{(\nu x)P \;\rightarrow\; (\nu x)P'}\,\text{(Res)}$$

$$\frac{Q \equiv P \qquad P \;\rightarrow\; P' \qquad P' \equiv Q'}{Q \;\rightarrow\; Q'}\,\text{(Struct)}$$

Equivalence of processes

- A sequential system is a function:
  inputs $\longrightarrow$ outputs
- Two functions are equivalent iff their outputs are identical for every input.

Equivalence of processes

- A sequential system is a function:
  inputs $\longrightarrow$ outputs
- Two functions are equivalent iff their outputs are identical for every input.
- A parallel system may not be deterministic.
- A parallel system may not terminate.

Traces

$$P \xrightarrow{a} P_1 \xrightarrow{\tau} P_2 \xrightarrow{\cdots} \ldots$$

is a trace of P

- $\xrightarrow{a}$: synchronization on channel a (for example $\bar{a}v \mid a(x).P$)
- $\xrightarrow{\tau}$: synchronization on an internal private channel (for example $(\nu a)(\bar{a}v \mid a(x).P)$)

Traces

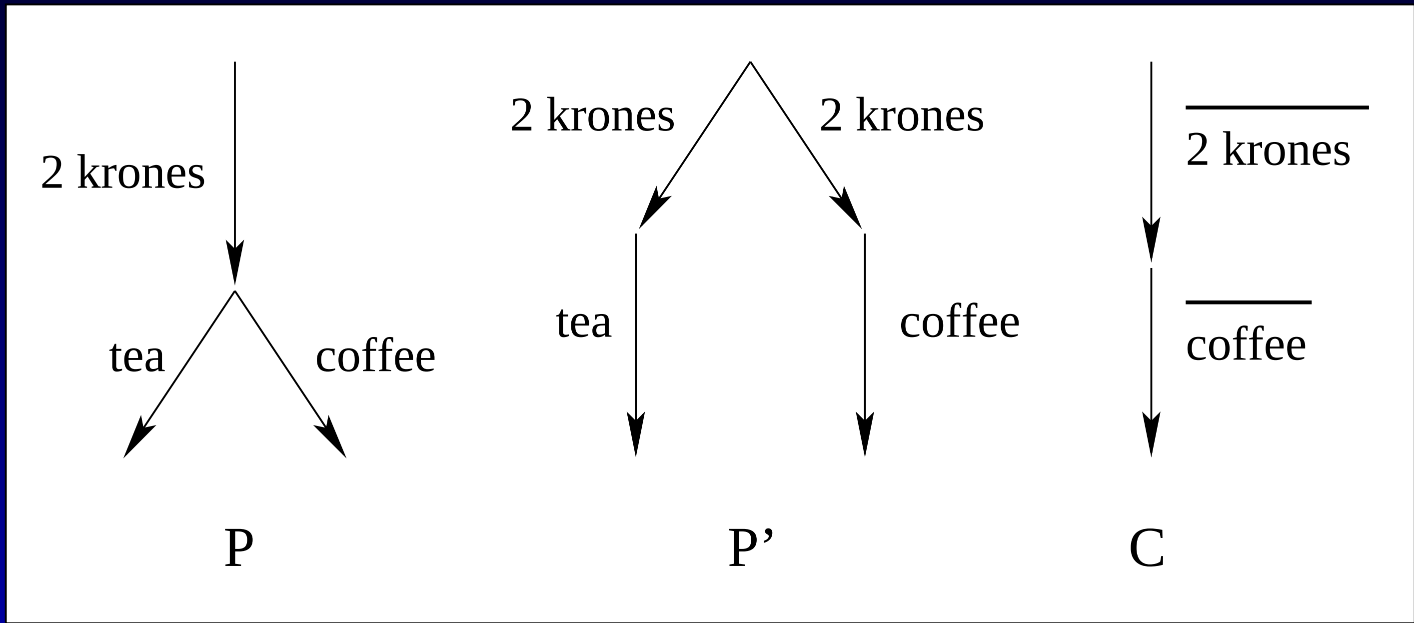
$$P \xrightarrow{a} P_1 \xrightarrow{\tau} P_2 \xrightarrow{\cdots} \ldots$$

is a trace of P

- $\xrightarrow{a}$: synchronization on channel a (for example $\bar{a}v \mid a(x).P$)
- $\xrightarrow{\tau}$: synchronization on an internal private channel (for example $(\nu a)(\bar{a}v \mid a(x).P)$)
- (informal) P equivalent to Q: same set of traces (maybe infinite)

Compositionality

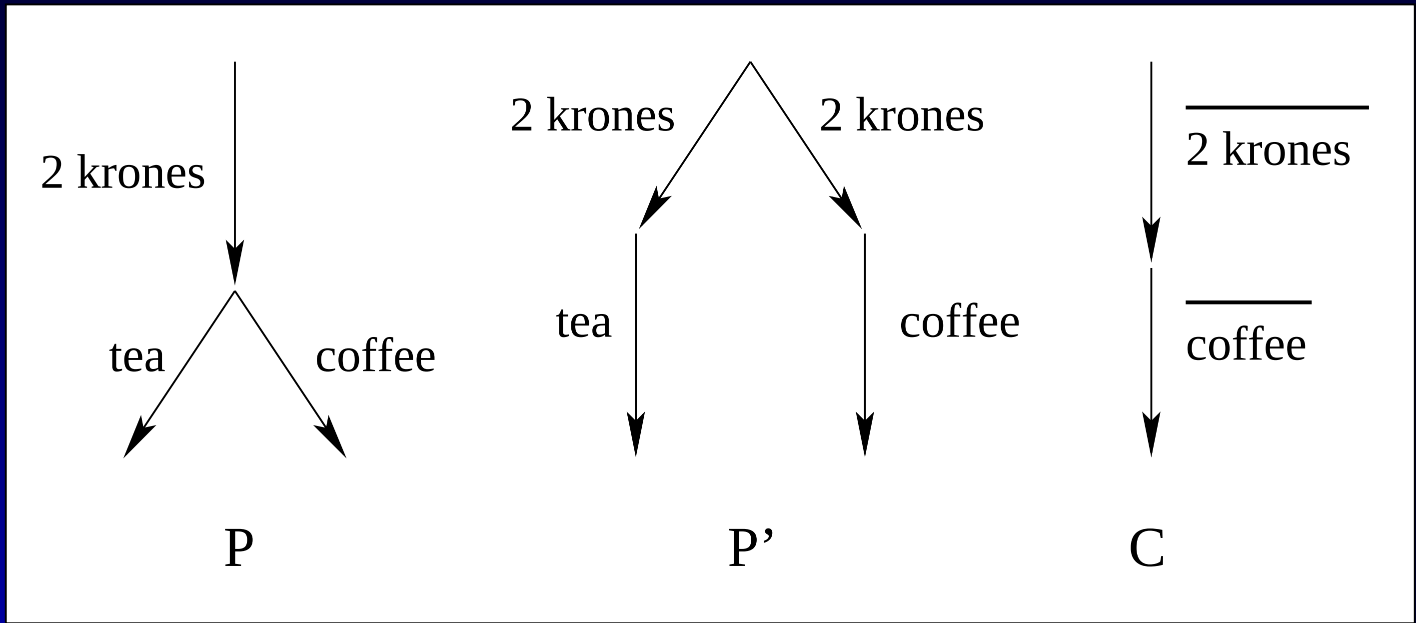
Two coffee machines and a consumer:



$$
\begin{aligned}
P &= 2\,krones.(tea + coffee) \\
P' &= (2\,krones.tea) + (2\,krones + coffee) \\
C &= \overline{2\,krones}\,.\,\overline{coffee}
\end{aligned}
$$

Compositionality

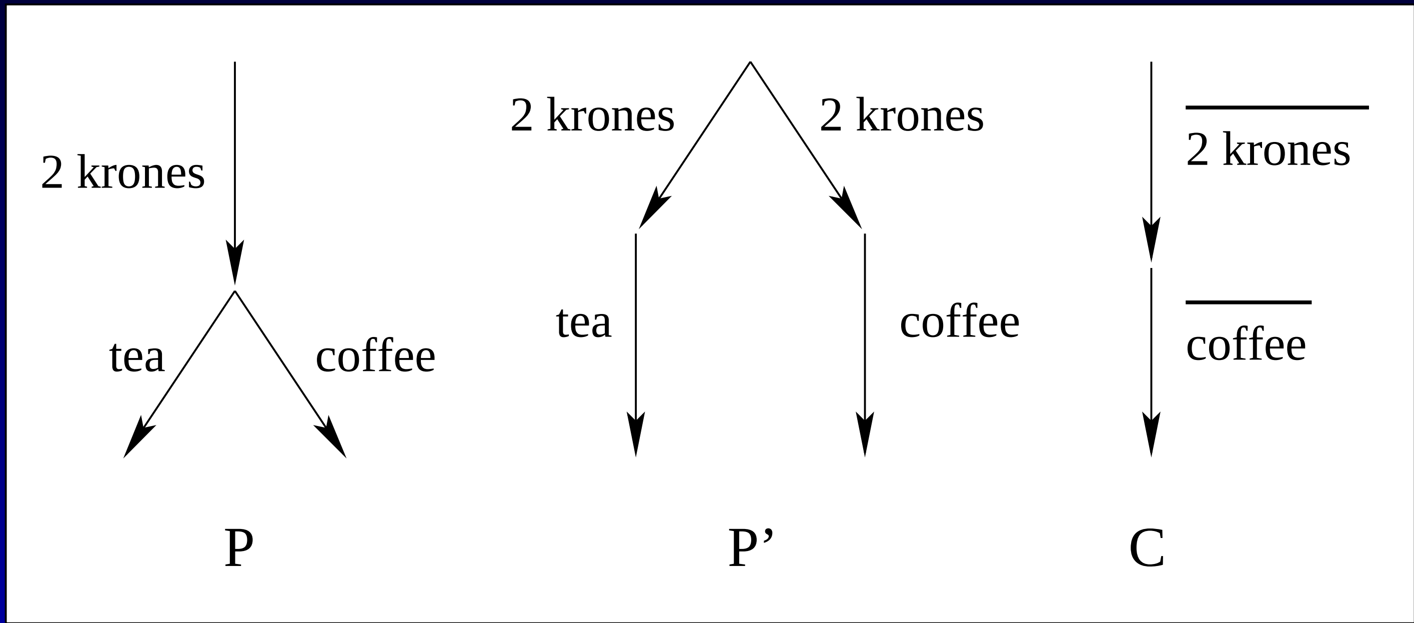
Two coffee machines and a consumer:



- P and P' accept the same language
- $P|C$ and $P'|C$ do not accept the same language

Compositionality

Two coffee machines and a consumer:



- P and P' accept the same language
- $P|C$ and $P'|C$ do not accept the same language
- $\Rightarrow$ trace equivalence is not compositional

Barbs

- Instead of looking at what happens, let's see what we are able to do (intensionality)
- Observing a state: $P \downarrow \eta$ if P contains a toplevel visible prefix whose subject is η (either a or $\bar{a}$)
- Remark: $P \downarrow a$ can be defined as

$$P \equiv (\nu\vec{n})((a(x).P' + M) \mid Q)$$

for some $\vec{n}$, P', M and Q such that $a \notin \vec{n}$

Bisimularity

Definition [Barbed bisimulation] A relation $\mathcal{R}$ is a *barbed bisimulation* iff $P\mathcal{R}Q$ implies $(\forall\eta.P \downarrow \eta \Rightarrow Q \downarrow \eta)$ and, for any P' such that $P \longrightarrow P'$, there is a process Q' such that $Q \longrightarrow Q'$ and $P'\mathcal{R}Q'$, and symmetrically for Q.

Bisimularity

Definition [Barbed bisimulation] A relation $\mathcal{R}$ is a *barbed bisimulation* iff $P\mathcal{R}Q$ implies $(\forall \eta . P \downarrow \eta \Rightarrow Q \downarrow \eta)$ and, for any P' such that $P \longrightarrow P'$, there is a process Q' such that $Q \longrightarrow Q'$ and $P'\mathcal{R}Q'$, and symmetrically for Q.

Definition [Barbed bisimilarity] The *barbed bisimilarity* is the greatest barbed bisimulation. We write $P \dot{\sim} Q$.

Proposition $\dot{\sim}$ is an equivalence relation.

Bisimularity

Definition [Barbed bisimulation] A relation $\mathcal{R}$ is a *barbed bisimulation* iff $P\mathcal{R}Q$ implies $(\forall \eta . P \downarrow \eta \Rightarrow Q \downarrow \eta)$ and, for any P' such that $P \longrightarrow P'$, there is a process Q' such that $Q \longrightarrow Q'$ and $P'\mathcal{R}Q'$, and symmetrically for Q.

Definition [Barbed bisimilarity] The *barbed bisimilarity* is the greatest barbed bisimulation. We write $P \dot{\sim} Q$.

Proposition $\dot{\sim}$ is an equivalence relation.

Remark: to prove $P \dot{\sim} Q$, find one bisimulation $\mathcal{R}$ such that $P\mathcal{R}Q$.

Example

Let:

$$\begin{aligned}\mathcal{R} \;=\{ \;& ((\nu z)(\bar{z}a \mid z(w).\bar{x}w), \tau.\bar{x}b), \\ & (P, Q) \;\;/\;\; P \equiv (\nu z)(\mathbf{0} \mid \bar{x}a), Q \equiv \bar{x}b\}\end{aligned}$$

$\mathcal{R}$ is a barbed bisimulation, thus in particular:

$$(\nu z)(\bar{z}a \mid z(w).\bar{x}w) \stackrel{\cdot}{\sim} \tau.\bar{x}b$$

Example

Let:

$$\begin{aligned}\mathcal{R} \;= \{ \;& ((\nu z)(\bar{z}a \mid z(w).\bar{x}w), \tau.\bar{x}b), \\ & (P, Q) \;\; / \;\; P \equiv (\nu z)(\mathbf{0} \mid \bar{x}a), Q \equiv \bar{x}b\}\end{aligned}$$

$\mathcal{R}$ is a barbed bisimulation, thus in particular:

$$(\nu z)(\bar{z}a \mid z(w).\bar{x}w) \stackrel{.}{\sim} \tau.\bar{x}b$$

$\stackrel{.}{\sim}$ is quite weak... and still not compositional !

Contexts

- A context is a term with a *hole*, written $[]$.

$$\begin{array}{rcl} C & ::= & \mathbf{0} \mid a(x).C \mid \bar{a}x.C \mid (C_1 | C_2) \\ & & \mid (\nu a)C \mid !C \mid C_1 + C_2 \mid [] \end{array}$$

- $C[P]$ is the process obtained by replacing the hole $[]$ with P.

Contexts

- A context is a term with a *hole*, written $[]$.

$$C \quad ::= \quad \mathbf{0} \mid a(x).C \mid \bar{a}x.C \mid (C_1|C_2)$$
$$\mid (\nu a)C \mid !C \mid C_1 + C_2 \mid []$$

- $C[P]$ is the process obtained by replacing the hole $[]$ with P.
- Non-receptive context: no occurrence of $[]$ under an input prefix.

Barbed congruence

Definition [Barbed congruence and equivalence]
The barbed congruence (resp. barbed equivalence), written $\simeq^C$ (resp. $\simeq$), is the greatest congruence (resp. non-receptive congruence) included in $\dot{\sim}$.

Barbed congruence

Definition [Barbed congruence and equivalence]
The barbed congruence (resp. barbed equivalence), written $\simeq^C$ (resp. $\simeq$), is the greatest congruence (resp. non-receptive congruence) included in $\dot{\sim}$.

Example $\bar{z} \mid a \quad \begin{array}{c}\simeq \\ \not\simeq^C\end{array} \quad \bar{z}.a + a.\bar{z}$

Barbed congruence

Definition [Barbed congruence and equivalence] The barbed congruence (resp. barbed equivalence), written $\simeq^C$ (resp. $\simeq$), is the greatest congruence (resp. non-receptive congruence) included in $\dot{\sim}$.

Example $$\bar{z} \mid a \quad \begin{array}{c}\simeq \\ \not\simeq^C\end{array} \quad \bar{z}.a + a.\bar{z}$$

but

$$\bar{z} \mid a \simeq^C \bar{z}.a + a.\bar{z} + [z = a]\tau$$

Barbed congruence

Definition [Barbed congruence and equivalence]
The barbed congruence (resp. barbed equivalence), written $\simeq^C$ (resp. $\simeq$), is the greatest congruence (resp. non-receptive congruence) included in $\dot{\sim}$.

Example $\bar{z} \mid a \quad \begin{matrix} \simeq \\ \not\simeq^C \end{matrix} \quad \bar{z}.a + a.\bar{z}$

but

$$\bar{z} \mid a \simeq^C \bar{z}.a + a.\bar{z} + [z = a]\tau$$

Characterization
$P \simeq Q$ iff for any R, $P \mid R \,\dot{\sim}\, Q \mid R$.

Some general laws

- Restriction

$$(\nu a)(a(x).P) \simeq^C \mathbf{0}$$
$$(\nu x)(x(y).P \mid \bar{w}z.Q) \simeq^C \bar{w}z.(\nu x)(x(y).P \mid Q)$$

if $x \neq w$ and $x \neq z$.

- Replication $\begin{cases} !(P \mid Q) \simeq^C !P \mid !Q \\ !!P \simeq^C !P \\ !(P + Q) \simeq^C !(P \mid Q) \\ ![a = b]P \simeq^C [a = b]!P \\ !\eta.P \not\simeq^C \eta.!P, \quad \eta \text{ prefix} \\ !(\nu x)P \not\simeq^C (\nu x)!P \end{cases}$

Summary

We have defined an equivalence:

- with good properties, including compositionality
- describing *behaviours*
- relying on *observations* ($P \downarrow \eta$)

Labelled transition system

- Changing the point of view: we now consider the interactions with the environment.
- Three kinds of transition: $\begin{cases} P \stackrel{a(b)}{\longrightarrow} Q \\ P \stackrel{\bar{a}b}{\longrightarrow} Q, P \stackrel{\bar{a}(b)}{\longrightarrow} Q \\ P \stackrel{\tau}{\longrightarrow} Q \end{cases}$
- names: $n(\mu)$

  bound names: $bn(\bar{a}(b)) = \{b\}$
  $bn(\mu) = \emptyset$ otherwise

Operational semantics 1

$$\frac{}{\bar{a}b.P \xrightarrow{\bar{a}b} P}\,(\text{OUT}) \qquad \frac{}{a(x).P \xrightarrow{a(v)} P\{x \mapsto v\}}\,(\text{INP})$$

$$\frac{P \xrightarrow{a(b)} P' \qquad Q \xrightarrow{\bar{a}b} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}\,(\text{COMM})$$

Operational semantics 1

$$\frac{}{\bar{a}b.P \xrightarrow{\bar{a}b} P}\text{(OUT)} \qquad \frac{}{a(x).P \xrightarrow{a(v)} P\{x \mapsto v\}}\text{(INP)}$$

$$\frac{P \xrightarrow{a(b)} P' \qquad Q \xrightarrow{\bar{a}b} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}\text{(COMM)}$$

$$\frac{P \xrightarrow{\mu} P'}{P \mid Q \xrightarrow{\mu} P' \mid Q}\text{(PAR)} \quad bn(\mu) \cap fn(Q) = \emptyset$$

$$\frac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P'}\text{(SUM)}$$

$+$ symmetrical rules !

Operational semantics 2

$$\frac{!P \mid P \xrightarrow{\mu} P'}{!P \xrightarrow{\mu} P'}\text{(BANG)}$$

$$\frac{P \xrightarrow{\mu} P'}{(\nu a)P \xrightarrow{\mu} (\nu a)P'}\text{(RES)} \quad a \notin n(\mu)$$

Operational semantics 2

$$\frac{!P \mid P \xrightarrow{\mu} P'}{!P \xrightarrow{\mu} P'}\text{(BANG)}$$

$$\frac{P \xrightarrow{\mu} P'}{(\nu a)P \xrightarrow{\mu} (\nu a)P'}\text{(RES)} \quad a \notin n(\mu)$$

$$\frac{P \xrightarrow{\bar{a}b} P'}{(\nu b)P \xrightarrow{\bar{a}(b)} P'}\text{(OPEN)} \quad a \neq b$$

$$\frac{P \xrightarrow{a(b)} P' \qquad Q \xrightarrow{\bar{a}(b)} Q'}{P \mid Q \xrightarrow{\tau} (\nu b)(P' \mid Q')}\text{(CLOSE)}$$

Example...

Bisimilarity – again

Definition [Bisimulation] A relation $\mathcal{R}$ is a *bisimulation* iff, whenever $P\mathcal{R}Q$ and $P \xrightarrow{\mu} P'$, there is a process Q' such that $Q \xrightarrow{\mu} Q'$ and $P'\mathcal{R}Q'$, and symmetrically for Q.

Bisimilarity – again

Definition [Bisimulation] A relation $\mathcal{R}$ is a *bisimulation* iff, whenever $P\mathcal{R}Q$ and $P \xrightarrow{\mu} P'$, there is a process Q' such that $Q \xrightarrow{\mu} Q'$ and $P'\mathcal{R}Q'$, and symmetrically for Q.

Definition [Bisimilarity] The bisimilarity, written $\sim$, is the greatest bisimulation.

Bisimilarity – again

Definition [Bisimulation] A relation $\mathcal{R}$ is a *bisimulation* iff, whenever $P\mathcal{R}Q$ and $P \xrightarrow{\mu} P'$, there is a process Q' such that $Q \xrightarrow{\mu} Q'$ and $P'\mathcal{R}Q'$, and symmetrically for Q.

Definition [Bisimilarity] The bisimilarity, written $\sim$, is the greatest bisimulation.

Definition [Full bisimilarity] $P \sim^C Q$ iff $P\sigma \sim Q\sigma$ for any substitution σ.

Bisimilarity – again

Definition [Bisimulation] A relation $\mathcal{R}$ is a *bisimulation* iff, whenever $P \mathcal{R} Q$ and $P \xrightarrow{\mu} P'$, there is a process Q' such that $Q \xrightarrow{\mu} Q'$ and $P' \mathcal{R} Q'$, and symmetrically for Q.

Definition [Bisimilarity] The bisimilarity, written $\sim$, is the greatest bisimulation.

Definition [Full bisimilarity] $P \sim^C Q$ iff $P\sigma \sim Q\sigma$ for any substitution σ.

Remark: $\sim$ implies trace equivalence.

Example

Let's consider:

$$\begin{aligned}\mathcal{R} \; = \{ \;\; & ((\nu z)(\bar{z}a \mid z(w).\bar{x}w), \tau.\bar{x}a), \\ & ((\nu z)(\mathbf{0} \mid \bar{x}a), \bar{x}a), \\ & ((\nu z)(\mathbf{0} \mid \mathbf{0}), \mathbf{0})\}\end{aligned}$$

$\mathcal{R}$ is a bisimulation, thus in particular:

$$(\nu z)(\bar{z}a \mid z(w).\bar{x}w) \sim \tau.\bar{x}a$$

$\Rightarrow$ smaller relation $\mathcal{R}$

Comparing the definitions

Theorem

$P \simeq Q$ iff $P \sim Q$, and $P \simeq^C Q$ iff $P \sim^C Q$.

Comparing the definitions

Theorem

$P \simeq Q$ iff $P \sim Q$, and $P \simeq^C Q$ iff $P \sim^C Q$.

Remarks:

- $\sim$ can be seen as a proof technique for $\simeq$
- $\sim$ allows to *derive* the laws for $\equiv$ (structure $\longrightarrow$ behaviour)

Chimic vs labelled transitions

- $\longrightarrow$ More natural, we work modulo α-conversion, AC of $|$ and $+$ and permutation of ν.
  Definition of equivalence: more "declarative", context plays an important role.
- $\xrightarrow{\mu}$ We work on trees, with the redex "on" the term.
  Interactions between the term and the context are built more deterministically.
  Simplier definition of equivalence.

Late variant

We have seen an *early* operational semantics:

$$\frac{}{a(x).P \xrightarrow{a(v)} P\{x \mapsto v\}}\text{(INP)} \qquad \frac{P \xrightarrow{a(b)} P' \qquad Q \xrightarrow{\bar{a}b} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}\text{(COMM)}$$

Late variant

We have seen an *early* operational semantics:

$$\frac{}{a(x).P \xrightarrow{a(v)} P\{x \mapsto v\}}\text{(INP)} \qquad \frac{P \xrightarrow{a(b)} P' \quad Q \xrightarrow{\bar{a}b} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}\text{(COMM)}$$

We can actually differ the application of substitution:

$$\frac{}{a(x).P \xrightarrow{a(x)} P}\text{(INP)} \qquad \frac{P \xrightarrow{a(x)} P' \quad Q \xrightarrow{\bar{a}b} Q'}{P \mid Q \xrightarrow{\tau} P'\{x \mapsto b\} \mid Q'}\text{(COMM)}$$

Late variant

Definition A symmetrical relation $\mathcal{R}$ is a late bisimulation iff, whenever $P \mathcal{R} Q$:

- if $P \xrightarrow{a(x)} P'$, there is a process Q' such that $Q \xrightarrow{a(x)} Q'$ and, for all b, $P'\{x \mapsto b\} \; \mathcal{R} \; Q'\{x \mapsto b\}$;
- if $P \xrightarrow{\mu} P'$ where μ is not an input, usual definition.

Late variant

Definition A symmetrical relation $\mathcal{R}$ is a late bisimulation iff, whenever $P\mathcal{R}Q$:

- if $P \stackrel{a(x)}{\longrightarrow} P'$, there is a process Q' such that $Q \stackrel{a(x)}{\longrightarrow} Q'$ and, for all b, $P'\{x \mapsto b\} \,\mathcal{R}\, Q'\{x \mapsto b\}$;
- if $P \stackrel{\mu}{\longrightarrow} P'$ where μ is not an input, usual definition.

Theorem $\sim_l \subsetneq \sim$

Counter-example: $P = x(z) + x(z).\bar{z}$
$Q = x(z) + x(z).\bar{z} + x(z).[z = y]\bar{z}$

Proof techniques

$$
\begin{array}{ccc}
P & \sim & Q \\
\mu \downarrow & & \downarrow \mu \\
P' & \sim & Q'
\end{array}
\qquad
\begin{array}{ccc}
P & \mathcal{R} & Q \\
\mu \downarrow & & \downarrow \mu \\
P' & \mathcal{R} & Q'
\end{array}
\qquad
\begin{array}{ccc}
P & \mathcal{R} & Q \\
\mu \downarrow & & \downarrow \mu \\
P' & \mathcal{F}(\mathcal{R}) & Q'
\end{array}
$$

bisimilarity   bisimulation   bisimulation up-to $\mathcal{F}$

Proof techniques

$$\begin{array}{ccc} P & \sim & Q \\ \mu\downarrow & & \downarrow\mu \\ P' & \sim & Q' \end{array} \qquad \begin{array}{ccc} P & \mathcal{R} & Q \\ \mu\downarrow & & \downarrow\mu \\ P' & \mathcal{R} & Q' \end{array} \qquad \begin{array}{ccc} P & \mathcal{R} & Q \\ \mu\downarrow & & \downarrow\mu \\ P' & \mathcal{F}(\mathcal{R}) & Q' \end{array}$$

bisimilarity   bisimulation   bisimulation up-to $\mathcal{F}$

For example, bisimulation up-to bisimilarity:

$$\begin{array}{ccc} P & \mathcal{R} & Q \\ \mu\downarrow & & \downarrow\mu \\ P' & \sim P_1 \, \mathcal{R} \, Q_1 \sim & Q' \end{array}$$

Reference: D. Sangiorgi, "*On the bisimulation proof technique*"

Weak transitions

- Two kinds of transitions:
  - $\xrightarrow{\mu}$ with $\mu \neq \tau$ and $\xrightarrow{\tau}$
  - visible transitions and internal transitions
  - interaction with the context and no internal computation

Weak transitions

- Two kinds of transitions:
  - $\xrightarrow{\mu}$ with $\mu \neq \tau$ and $\xrightarrow{\tau}$
  - visible transitions and internal transitions
  - interaction with the context and no internal computation
- Idea: ignore the internal transitions
  $\longrightarrow$ *weak* equivalences

Weak transitions

- Two kinds of transitions:
  - $\xrightarrow{\mu}$ with $\mu \neq \tau$ and $\xrightarrow{\tau}$
  - visible transitions and internal transitions
  - interaction with the context and no internal computation
- Idea: ignore the internal transitions
  $\longrightarrow$ *weak* equivalences
- **Definition [weak transitions]**
  $\Rightarrow$: reflexive and transitive closure of $\xrightarrow{\tau}$
  $\xrightarrow{\hat{\mu}}$: $\xrightarrow{\tau}$ or $=$ when $\mu = \tau$, $\xrightarrow{\mu}$ otherwise
  $P \overset{\hat{\mu}}{\Rightarrow} P'$: $P \Rightarrow\xrightarrow{\hat{\mu}}\Rightarrow P'$

Weak bisimilarity

We play the game of bisimulation, changing the notion of "step":

Definition A relation $\mathcal{R}$ is a *weak bisimulation* iff,

> whenever $P \mathcal{R} Q$ and $P \stackrel{\hat{\mu}}{\Rightarrow} P'$, there is a process Q' such that $Q \stackrel{\hat{\mu}}{\Rightarrow} Q'$ and $P' \mathcal{R} Q'$, and symmetrically for Q. The *weak bisimilarity* is written $\approx$.

Weak bisimilarity

- $\approx$ is an equivalence relation
- $\sim \subseteq \approx$
- Some examples of laws:

$$\begin{array}{rcl} \alpha.\tau.P & \approx & \alpha.P \\ \tau.P & \approx & P \\ P + \tau.P & \approx & P \\ \alpha.(P + \tau.Q) + \alpha.Q & \approx & \alpha.(P + \tau.Q) \end{array}$$

- Also a presentation with barbs:

$$\Downarrow \eta \stackrel{def}{=} \Rightarrow \downarrow \eta$$

Asynchronous π

Only form of output: $\bar{a}b$

$$\begin{array}{rcl} P & ::= & \bar{x}y \mid M \mid P_1|P_2 \mid (\nu x)P \mid !P \\ M & ::= & \mathbf{0} \mid x(z).P \mid \tau.P \mid M + M' \end{array}$$

- More realistic
- Remark: $\tau.P$ and $\mathbf{0}$ can be encoded
- A choice $+$ hides some protocol
- Why no output in sums ?

Asynchrony

- No continuation for outputs, but there can be some causality relations:

$$(\nu y, z)(\bar{x}y \mid \bar{y}z \mid \bar{z}a \mid R) \quad \text{with } y, z \notin fn(R)$$

Asynchrony

- No continuation for outputs, but there can be some causality relations:

$$(\nu y, z)(\bar{x}y \mid \bar{y}z \mid \bar{z}a \mid R) \quad \text{with } y, z \notin fn(R)$$

- If $P \stackrel{\bar{x}y}{\rightarrow} P'$, then $P \equiv \bar{x}y \mid P'$

Asynchrony

- No continuation for outputs, but there can be some causality relations:

$$(\nu y, z)(\bar{x}y \mid \bar{y}z \mid \bar{z}a \mid R) \quad \text{with } y, z \notin fn(R)$$

- If $P \xrightarrow{\bar{x}y} P'$, then $P \equiv \bar{x}y \mid P'$
- If $P \xrightarrow{\bar{x}(y)} P'$, then $P \equiv (\nu y)(\bar{x}y \mid P')$

Asynchrony

- No continuation for outputs, but there can be some causality relations:

$$(\nu y, z)(\bar{x}y \mid \bar{y}z \mid \bar{z}a \mid R) \quad \text{with } y, z \notin fn(R)$$

- If $P \xrightarrow{\bar{x}y} P'$, then $P \equiv \bar{x}y \mid P'$
- If $P \xrightarrow{\bar{x}(y)} P'$, then $P \equiv (\nu y)(\bar{x}y \mid P')$
- If $P \xrightarrow{\bar{x}y}\xrightarrow{\mu} P'$, then $P \xrightarrow{\mu}\xrightarrow{\bar{x}y}\equiv P'$ (confluence)

Asynchrony

- No continuation for outputs, but there can be some causality relations:

$$(\nu y, z)(\bar{x}y \mid \bar{y}z \mid \bar{z}a \mid R) \quad \text{with } y, z \notin fn(R)$$

- If $P \xrightarrow{\bar{x}y} P'$, then $P \equiv \bar{x}y \mid P'$
- If $P \xrightarrow{\bar{x}(y)} P'$, then $P \equiv (\nu y)(\bar{x}y \mid P')$
- If $P \xrightarrow{\bar{x}y}\xrightarrow{\mu} P'$, then $P \xrightarrow{\mu}\xrightarrow{\bar{x}y}\equiv P'$ (confluence)
- If $P \xrightarrow{\bar{x}y}\xrightarrow{x(w)} P'$ with $w \notin fn(P)$, then $P \xrightarrow{\tau}\equiv P'\{w \mapsto y\}$

Asynchrony

Theorem The notions of early and late bisimulations coincide in asynchronous π-calculus. Moreover, these are congruences.

$\Rightarrow$ a simpler theory, easier proofs...

Encodings

Notation: encoding of P: $[\![P]\!]$

Interest:

- to compare models, programming paradigms and idioms
- to study the expressive power of a construction and subfragments of a language

Encodings

- We want to show something like $\forall P.P \asymp [\![P]\!]$ where $\asymp$ is some notion of equivalence (weak/strong bisimilarity, trace equivalence...).

Encodings

- We want to show something like $\forall P. P \asymp [\![P]\!]$ where $\asymp$ is some notion of equivalence (weak/strong bisimilarity, trace equivalence...).
- This makes sense only when $[\![P]\!]$ and P are in a same language. Often, we use $\approx$ (encoding a construction into a smaller language).

Encodings

- We want to show something like $\forall P.P \asymp [\![P]\!]$ where $\asymp$ is some notion of equivalence (weak/strong bisimilarity, trace equivalence...).
- This makes sense only when $[\![P]\!]$ and P are in a same language. Often, we use $\approx$ (encoding a construction into a smaller language).
- Otherwise, we might want to prove *full abstraction*:

$$P_1 \asymp P_2 \quad \text{iff} \quad [\![P_1]\!] \asymp [\![P_2]\!]$$

(allows to compare encodings from one language into another)

Encodings

Otherwise, we shall prove at least *operational correspondence*:

- If $P \longrightarrow P'$, then $[\![P]\!] \longrightarrow [\![P']\!]$.
- If $[\![P]\!] \longrightarrow Q$, then there is a process P' such that $P \longrightarrow P'$ and $Q \equiv [\![P']\!]$.

(one-to-one version)

Encodings

Otherwise, we shall prove at least *operational correspondence*:

- If $P \rightarrow P'$, then $[\![P]\!] \Rightarrow [\![P']\!]$.
- If $[\![P]\!] \Rightarrow Q$, then there is a process P' such that $P \rightarrow P'$ and $Q \equiv [\![P']\!]$.

(weak version)

Encodings

Otherwise, we shall prove at least *operational correspondence*:

- If $P \rightarrow P'$, then $[\![P]\!] \Rightarrow\approx [\![P']\!]$.
- If $[\![P]\!] \Rightarrow Q$, then there is a process P' such that $P \rightarrow P'$ and $Q \approx [\![P']\!]$.

(weak version up-to bisimilarity)

Encoding synchronous π

How should we represent $\bar{a}v.P \mid a(x).Q$ in asynchronous π-calculus ?

Encoding synchronous π

How should we represent $\bar{a}v.P \mid a(x).Q$ in asynchronous π-calculus ?

$$(\nu t)(\bar{a}\langle v, t\rangle \mid t.P) \mid a(x, r).(Q \mid \bar{r})$$

Encoding synchronous π

How should we represent $\bar{a}v.P \mid a(x).Q$ in asynchronous π-calculus ?

$$(\nu t)(\bar{a}\langle v, t\rangle \mid t.P) \mid a(x, r).(Q \mid \bar{r})$$

- One can show that $[\![P]\!] \approx [\![Q]\!]$ implies $P \approx Q$.

Encoding synchronous π

How should we represent $\bar{a}v.P \mid a(x).Q$ in asynchronous π-calculus ?

$$(\nu t)(\bar{a}\langle v, t\rangle \mid t.P) \mid a(x, r).(Q \mid \bar{r})$$

- One can show that $[\![P]\!] \approx [\![Q]\!]$ implies $P \approx Q$.
- $\Leftarrow$??

Encoding synchronous π

How should we represent $\bar{a}v.P \mid a(x).Q$ in asynchronous π-calculus ?

$$(\nu t)(\bar{a}\langle v, t\rangle \mid t.P) \mid a(x, r).(Q \mid \bar{r})$$

- One can show that $[\![P]\!] \approx [\![Q]\!]$ implies $P \approx Q$.
- $\Leftarrow$??

Take $A \stackrel{def}{=} \bar{a}v.\bar{a}v$ and $B \stackrel{def}{=} \bar{a}v \mid \bar{a}v$
We have $A \sim B$, but:

$$\begin{array}{rrcl} & [\![A]\!] & \equiv & (\nu t_1, t_2)(\bar{a}\langle v, t_1\rangle \mid t_1.(\bar{a}\langle v, t_2\rangle \mid t_2)) \\ \text{and} & [\![B]\!] & \equiv & (\nu t_1)(\bar{a}\langle v, t_1\rangle \mid t_1) \mid (\nu t_2)(\bar{a}\langle v, t_2\rangle \mid t_2) \end{array}$$

Asynchronous π

Palamidessi, 1997

- Impossible to encode synchronous π into asynchronous π (with a *reasonable* encoding).

Asynchronous π

Palamidessi, 1997

- Impossible to encode synchronous π into asynchronous π (with a *reasonable* encoding).
- Because of mixed choice

$$a(x).P + \bar{b}v.Q$$

- Proof: impossible to resolve the problem of chief election in a symmetrical network.

Asynchronous π

Palamidessi, 1997

- Impossible to encode synchronous π into asynchronous π (with a *reasonable* encoding).
- Because of mixed choice

$$a(x).P + \bar{b}v.Q$$

- Proof: impossible to resolve the problem of chief election in a symmetrical network.
- "Reasonable" means: compositional ($[\![P|Q]\!] = [\![P]\!]|[\![Q]\!], [\![P\sigma]\!] = [\![P]\!]\sigma$) preserving divergence
- one of the very few non-expressivity result

λ-calculus

Terms:

$$M ::= x \mid \lambda x.M \mid (M\ M')$$

β-reduction:

$$(\lambda x.M)\ N \longrightarrow M\{x \mapsto N\}$$

λ-calculus

Terms:

$$M ::= x \mid \lambda x.M \mid (M\ M')$$

β-reduction:

$$(\lambda x.M)\ N \rightarrow M\{x \mapsto N\}$$

Encoding the λ-calculus into π, ideas:

- A λ-term M is represented by a π-term $[\![M]\!]$ located in p: $[\![M]\!]_p$.
- Application is represented with parallel composition.

Encoding the λ-calculus

$$
\begin{aligned}
[\![\lambda x.M]\!]_p &\stackrel{def}{=} (\nu y)\bar{p}y.!y(x,q).[\![M]\!]_q \\
[\![x]\!]_p &\stackrel{def}{=} \bar{p}x \\
[\![M\ N]\!]_p &\stackrel{def}{=} (\nu q)([\![M]\!]_q \mid q(v). \\
&\qquad\qquad (\nu r)([\![N]\!]_r \mid r(v').\bar{v}\langle v', p\rangle))
\end{aligned}
$$

Encoding the λ-calculus

$$
\begin{aligned}
[\![\lambda x.M]\!]_p &\stackrel{def}{=} (\nu y)\bar{p}y.!y(x,q).[\![M]\!]_q \\
[\![x]\!]_p &\stackrel{def}{=} \bar{p}x \\
[\![M\ N]\!]_p &\stackrel{def}{=} (\nu q)([\![M]\!]_q \mid q(v). \\
&\qquad\qquad (\nu r)([\![N]\!]_r \mid r(v').\bar{v}\langle v', p\rangle))
\end{aligned}
$$

- $[\![M]\!]_p$ sends the value of M on p
- For a function, we send its address;
  it is consulted by sending a value and a return channel.