The π-calculus and its family: interaction, equivalences

Davide Sangiorgi

INRIA Sophia-Antipolis

davide.sangiorgi@inria.fr
http://www-sop.inria.fr/meije/personnel/Davide.Sangiorgi.html

April 00

The π-calculus

$a, b, \ldots, x, y, z, \ldots$ Names

Processes

P	$::=$	$\mathbf{0}$	nil process
	$\mid$	$P \mid P$	parallel
	$\mid$	$\boldsymbol{\nu} a\, P$	restriction
	$\mid$	$a(b).\, P$	input
	$\mid$	$\overline{a}\, b\, .\, P$	output
	$\mid$	$!P$	replication
	$\mid$	$P + P$	summation
	$\mid$	$[a = b]P$	matching

Abbreviations:

$\overline{a}(b).\, P$ for $\boldsymbol{\nu} b\, \overline{a}\, b\, .\, P$

$\overline{a}.\, P$ and $a.\, P$ when the name communicated is not important

α for $\alpha.\, \mathbf{0}$, where α is any prefix

Behavioural equivalences (bisimilarity)

$P \sim Q$ implies

– If $P \xrightarrow{a(x)} P'$ then $\exists\, Q'$ s.t. $Q \xrightarrow{a(x)} Q'$ and $\forall\, b \quad P'\{b/x\} \sim Q'\{b/x\}$;

– if $P \xrightarrow{\mu} P'$ (μ not an input) then $\exists\, Q'$ s.t. $Q \xrightarrow{\mu} Q'$ and $P' \sim Q'$.

Interactions involve arbitrary substitutions among names.

$$\overline{a}\, b\,.\, P \mid \underbrace{a(y).\boxed{\begin{matrix} \ldots y \ldots \\ b \ldots y \ldots b \end{matrix}}}_{a(y).\,Q} \quad \xrightarrow{\tau} \quad P \mid \underbrace{\boxed{\begin{matrix} \ldots b \ldots \\ b \ldots b \ldots b \end{matrix}}}_{Q\{b/x\}}$$

aliasing ... (sigh!)

This talk

- $\Rightarrow$ relationship between interaction and substitution/aliasing
- ... in the π family (= π-calculus and its subcalculi)

Unfortunately...

$$\boxed{P \sim Q \quad \not\Rightarrow \quad P\sigma \sim Q\sigma}$$

Some consequences:

– very heavy case analysis in proofs (exponential on # free names)

– bisimilarity is not a congruence

– various bisimilarity relations

– axiomatisations $\left\{ \begin{array}{l} \text{hard (but possible)} \\ \text{no canonical normal forms (open problem)} \end{array} \right.$

Examples:

- $$[a = b]\overline{c} \quad \sim \quad \mathbf{0}$$

  but

  $$\begin{array}{ccc} ([a = b]\overline{c})\{^{b}\!/\!_{a}\} & \not\sim & \mathbf{0}\{^{b}\!/\!_{a}\} \\ \| & & \| \\ [b = b]\overline{c} & & \mathbf{0} \end{array}$$

- $$\overline{x} \mid y \quad \sim \quad \overline{x}.\, y + y.\, \overline{x}$$

  but

  $$\begin{array}{ccc} (\overline{x} \mid y)\{^{x}\!/\!_{y}\} & \not\sim & (\overline{x}.\, y + y.\, \overline{x})\{^{x}\!/\!_{y}\} \\ \| & & \| \\ \overline{x} \mid x & & \overline{x}.\, x + x.\, \overline{x} \\ \downarrow \tau & & \downarrow \tau \\ \mathbf{0} & & ?? \end{array}$$

Another problem of "observables" related to aliasing:

the notion of observable is not robust

(especially important in typed π-calculi)

I will not discuss it in this talk

Solution 1: refine the bisimilarity

solution 1.a: causality

(assume no matching, for simplicity)

$$\overline{x}.\, y + y.\, \overline{x} \quad \xrightarrow[0]{\overline{x}} \quad y \quad \xrightarrow[-1]{y} \quad \mathbf{0}$$

$$\overline{x} \mid y \quad \xrightarrow[0]{\overline{x}} \quad \mathbf{0} \mid y \quad \xrightarrow[0]{y} \quad \mathbf{0}$$

Theorem [Boreale and Sangiorgi, 94]
(i) Causal bisimilarity is preserved by substitutions
(ii) Causal bisimilarity is a congruence relation

Drawbacks:

– more complex definition and theory

– causality is a "vague" notion

For instance, what are the causal relations in

$$\boldsymbol{\nu} b\, (\overline{a}\, b \mid \overline{a}\, b \mid b(x).\, P) \qquad ?$$

solution 1.b: open bisimilarity

$\overline{x}.\, y + y.\, \overline{x} \quad \xrightarrow{??} \quad \mathbf{0} \qquad\qquad\qquad \overline{x} \mid y \quad \xrightarrow{\tau,\ \{x=y\}} \quad \mathbf{0}$

Theorem [Sangiorgi, 93]
(i) Open bisimilarity is preserved by substitutions
(ii) Open bisimilarity is a congruence relation

Also, an axiomatisation simpler than that of standard bisimilarity

Drawbacks:

– more complex definition

– too fine (true??)

– no mismatching (problem??)

Solution 2: cut the calculus

(a) remove matching

(b) remove summation

- You may want to remove them anyway

- Drawbacks:
  - algebraic theory
  - expressiveness:
    - matching [??] (does matching increase the expressiveness?)
    - summation [Palamidessi 97, Nestmann 96,98]

.... not quite enough (still $P \sim Q \not\Rightarrow P\sigma \sim Q\sigma$)

Counterexample [Boreale and Sangiorgi, 94] (weak bisimilarity):

$$
\begin{array}{lcl}
P & \stackrel{\text{def}}{=} & !\boldsymbol{\nu} d\,(\overline{z}\,d\,.\,x(e).\,\overline{a}(h).\,\overline{h}\,d\,.\,\overline{h}\,e\,) \mid\, !\boldsymbol{\nu} d\,(x(e).\,\overline{z}\,d\,.\,\overline{a}(h).\,\overline{h}\,d\,\overline{h}\,e\,) \\
Q & \stackrel{\text{def}}{=} & !(\boldsymbol{\nu} d, b\,)(\overline{z}\,d\,.\,\overline{b} \mid x(e).\,b.\,\overline{a}(h).\,\overline{h}\,d\,.\,\overline{h}\,e\,)
\end{array}
$$

Any simpler counterexamples?

Any easy fix? (for instance: does the property holds on finite processes?)

For now, we have to remove something else...

(c) Remove output prefix

⇒ **asynchronous π-calculus** (Aπ)

[Honda and Tokoro 91, Boudol 93]

$$a(y).\,Q \mid \overline{a}\,b \xrightarrow{\tau} Q\{^{b}\!/\!_{y}\}$$

- easier to implement
- a "functional" programming style

Theorem [Honda 92, Sangiorgi 94] In Aπ:
(i) bisimilarity is preserved by substitutions
(ii) bisimilarity is a congruence relation.

Thus the input clause of bisimilarity can be simplified:

If $P \xrightarrow{a(x)} P'$ then $\exists\ Q'$ s.t. $Q \xrightarrow{a(x)} Q'$ and $P' \sim Q'$

(rathen than: .... $\forall\, b \quad P'\{^{b}\!/\!_{x}\} \sim Q'\{^{b}\!/\!_{x}\}$)

Drawbacks

– algebraic theory even weaker

– observables non-robust

– expressiveness (??)

Theorem There is no compositional fully abstract encoding of $\pi_{\{=,+\}}$ into Aπ.

Proof: By the previous counterexamples and theorems. ■

(d) Disallow inputs of x in the body P of $a(x).\,P$

$\Rightarrow$ **locality**, LAπ

[cf: Join calculus; Merro and Sangiorgi 98]

$$\overbrace{\boldsymbol{\nu}a\ \boxed{\begin{array}{c} \ldots a(y)\ldots \\ a(z)\ldots\overline{a}\,v\ \ldots \end{array} \mid \overline{b}\,a}}^{\boldsymbol{\nu}a\,(P\,\mid\,\overline{b}\,a)} \qquad\mid\qquad b(x).\overbrace{\boxed{\begin{array}{c}\ldots\overline{x}\,w\ \ldots\\ \ldots\overline{c}\,x\ \ldots\overline{x}\,u\end{array}}}^{Q}$$

$$\downarrow \tau$$

$$\boldsymbol{\nu}a\ \Big(\ \underbrace{\boxed{\begin{array}{c} \ldots a(y)\ldots \\ a(z)\ldots\overline{a}\,v\ \ldots \end{array}}}_{P} \qquad\mid\qquad \underbrace{\boxed{\begin{array}{c}\ldots\overline{a}\,w\ \ldots\\ \ldots\overline{c}\,a\ \ldots\overline{a}\,u\end{array}}}_{Q\{a/x\}}\ \Big)$$

You may think of P as an **object**, named a
(better: one P's methods is named a)

- an **“object-oriented”** programming style
  (a bit strong as a claim, maybe; yet in the π semantics of objects [Walker, Sangiorgi-Kleist, Huttel-Kleist-Nestmann-Merro] locality – possibly dissociated with asynchrony – is essential)

- **algebraic theory** (richer than Aπ, still less rich than π)

- robust **observables**

- **type inference** à la ML

Drawbacks

- **expressiveness** (??)

- standard **bisimilarity** too strong (can be adjusted, maintaining the congruence property, but the definition becomes more complex)

Examples of algebraic laws

write $\approx_{L\pi}$ for bisimilarity in L$\mathrm{A}\pi$

$\mathsf{def}\ x(y) = R\ \mathsf{in}\ P$ for $\boldsymbol{\nu} x\,(!x(y).\,R \mid P)$

(a distributivity law for local environments)

$$\begin{array}{rll} \mathsf{def}\ x(y) = R\ \mathsf{in}\ (P \mid Q) & \approx_{L\pi} & \mathsf{def}\ x(y) = R\ \mathsf{in}\ P \\ & \mid & \mathsf{def}\ x(y) = R\ \mathsf{in}\ Q \end{array}$$

write $z \rhd x$ for $!z(a).\,\overline{x}\,a$ *(a link)*

(from external to internal mobility)

$$\overline{y}\,x \quad \approx_{L\pi} \quad \boldsymbol{\nu} z\,(\overline{y}\,z \mid z \rhd x)$$

where z is fresh

(a′) replace free output, $\overline{a}\, b\,.\, P$ with bound output, $\overline{a}(b).\, P \stackrel{\text{def}}{=} \boldsymbol{\nu} b\, (\overline{a}\, b\,.\, P)$

$\Rightarrow$ Internal π (or Private π), Iπ [Sangiorgi, 95]

The calculus Iπ

$$
\begin{array}{llll}
 & & & \text{Processes} \\
P & ::= & \mathbf{0} & \text{nil process} \\
 & \mid & P \mid P & \text{parallel} \\
 & \mid & \boldsymbol{\nu} a\, P & \text{restriction} \\
 & \mid & a(b).\, P & \text{input} \\
 & \mid & \overline{a}(b).\, P & \text{output} \\
 & \mid & P + P & \text{summation} \\
 & \mid & \ldots & \text{recursion}
\end{array}
$$

- Interaction:

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

- In Iπ there is *internal*, but not external, mobility.

Symmetries and dualities

Every process P has a dual $\overline{P}$ s.t.

$$\begin{array}{ccc} P & \leftrightarrow & \overline{P} \\ \mu \downarrow & & \overline{\mu} \downarrow \\ P' & \leftrightarrow & \overline{P'} \end{array}$$

Iπ^{-}: We strengthen the duality input/output by requiring that

1. in an **input** $x(y).\, P$, the object y may only be used in **output** in P;

2. in an **output** $\overline{x}(y).\, P$, the object y may only be used in **input** in P.

Example: $a(b).\, \overline{b}(c).\, c(d).\, \mathbf{0} \mid \overline{a}(b).\, b(c).\, \overline{c}(d).\, \mathbf{0}$

- In π, or Iπ, we can write:

$$a(y).\,\overline{y}(z).\,C \mid \overline{a}(x).\,(x(c).\,A \mid \overline{x}(y).\,B)$$

  There can be interferences between $\overline{y}(z).\,C$ and $\overline{x}(y).\,B$.

- By contrast, in any Iπ^- process $x(y).\,P \mid \overline{x}(y).\,Q$:

– all interferences on **outputs** at y are local to P;

– all interferences on **inputs** at x are local to Q.

[syntactic control of interferences]
can Reynold et al. 's work help?

- symmetry (syntax and semantics)

- operational theory: CCS plus alpha conversion

  This holds for:
  - SOS semantics
  - bisimilarity
  - axiomatisation
  - canonical normal forms.

- robust observables

Encoding (core) π-calculus into Iπ

$$
\left.
\begin{array}{ll}
\pi & \\
\downarrow & \text{Honda and Tokoro [1991], Boudol [1992]} \\
\mathsf{A}\pi & \\
\downarrow & \text{Boreale [1996]}
\end{array}
\right\}\ \text{folklore?}
$$

$$
\begin{array}{ll}
\mathsf{LA}\pi & \\
\downarrow & \text{Boreale [1996], Merro [1999]} \\
\mathsf{I}\pi^{-} &
\end{array}
$$

Endoding LAπ into Iπ^-

$$[\![\overline{a}\, b]\!] \stackrel{\text{def}}{=} \overline{a}(c).\, c \multimap b$$

where $\quad c \multimap b \stackrel{\text{def}}{=} !c(x).\, \overline{b}(y).\, y \multimap x$

Example:

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

$$\begin{array}{rcl}
[\![\overline{a}\, b \mid a(x).\, \overline{x}\, v\,]\!] & \stackrel{\text{def}}{=} & \\
\overline{a}(c).\, c \multimap b \mid a(x).\, \overline{x}(w).\, w \multimap v & \xrightarrow{\tau} & \\
\boldsymbol{\nu} c\, (c \multimap b \mid \overline{c}(w).\, w \multimap v) & \xrightarrow{\tau} & \\
(\boldsymbol{\nu} c, w\,)(\overline{b}(u).\, u \multimap w \mid c \multimap b \mid w \multimap v) & \sim & \\
(\boldsymbol{\nu} w\,)(\overline{b}(u).\, u \multimap w \mid w \multimap v) & \sim & \\
\overline{b}(u).\, (\boldsymbol{\nu} w\,)(u \multimap w \mid w \multimap v) & \approx & \\
\overline{b}(u).\, u \multimap v & = & [\![\overline{b}\, v\,]\!]
\end{array}$$

A useful law: $\boldsymbol{\nu} b\, (a \multimap b \mid b \multimap c) \approx a \multimap c$.

NB: strong similarities with the implementation of free outputs in the Pict compiler.

Theorem [Merro, 1999] Let $P, Q \in$ LAπ.
P and Q are asynchronously barbed congruent in LAπ

iff

$[\![P]\!]$ and $[\![Q]\!]$ are asynchronously barbed congruent in the asynchronous Iπ^-.

NB: In CCS or Iπ barbed congruence coincides with weak bisimulation; we have similar characterisations for the asynchronous barbed congruence (on asynchronous calculi).

Is the encoding really satisfactory?

Can Iπ and π (better: LAπ) be separated?

Theory of Iπ^-?

Conclusions

I  π