The linear π-calculus

Mauro Piccolo

February 16,2007

References

VY06 D. Varacca , N. Yoshida *Typed Event structure and the π-calculus* In MFPS, 2006

YHB03 N. Yoshida, K. Honda, M. Berger *Strong Normalization and the π-calculus* Journal of Information and Computation, 2003

YHB01 N. Yoshida, K. Honda, M. Berger *Sequentiality and the π-calculus* In Proc. of TLCA'01, 2001

San95 D. Sangiorgi *Internal Mobility and Agent passing calculi* In Proc. ICALP'95, 1995

SW01 D. Sangiorgi , D. Walker *The π-calculus: a Theory of Mobile Processes* Cambridge University Press, 2001

Bor98 M. Boreale *On the expressivness of Internal Mobility in Name-Passing calculi* Theoretical Computer Science, 1998

Introduction

Concurrent system

- a set of processes (agents) that interact togheter
  - concurrent computation
  - synchronization
  - mobility
- each process is defined by specifying all its possible interaction with the rest of the system

How do we describe a such complex reality?

The π-calculus

Names and variables:

a, b, c (names) , $x, y, z,$ (var.) s, t(both)

Action capabilities:

$\pi ::=$	$a(\vec{x})$	input
	$\overline{a}\langle\vec{t}\rangle$	output
	τ	internal action

Processes:

$P ::=$	$\pi.P$	prefixed action
	$\sum_{i\in I} \pi_i.P_i$	prefixed sum
	$P\vert P$	parallel composition
	$\nu x P$	restriction
	$!P$	replication
	0	termination

$a(x).P$ and $\nu x P$ are binding constructs (the occurences of x are bounded in P)

Examples:

input and output actions

$$P = a(x).\overline{x}\langle t\rangle.P'$$

parallel execution

$$Q = a(x).Q_1|\overline{v}(t).Q_2$$

possible interaction

$$S = a(x).S|\overline{a}\langle t\rangle.S'$$

restriction (scope extrusion)

$$R = \nu z(\overline{u}\langle z\rangle.z(w).R')$$

Operational semantics

Behaviour of a process

- internal dynamics of a process
  - reduction rules
- interaction with the environmet
  - labelled transition system

The **structural congruence** ($\equiv$) is the smallest contextual congruence satisfying the following axioms

α-conversion of bound names	$P\|(Q\|R) \equiv (P\|Q)\|R$
$P\|Q \equiv Q\|P$	$P\|0 \equiv P$
$\nu x \nu y P \equiv \nu y \nu x P$	$\nu z 0 \equiv 0$
$\nu z(P\|Q) \equiv (\nu z P)\|Q$ if $z \notin fn(Q)$	$!P \equiv P\|!P$

The **reduction** ($\longrightarrow$) is defined by the following rules

$$a(\vec{x}).P + M|\overline{a}\langle\vec{t}\rangle.Q + N \longrightarrow P\{\vec{t}/\vec{x}\}|Q \quad (1)$$

$$\tau.P + M \longrightarrow P \quad (2)$$

$$P \longrightarrow P' \Rightarrow P|Q \longrightarrow P'|Q \quad (3)$$

$$P_1 \equiv Q_1 \longrightarrow Q_2 \equiv P_2 \Rightarrow P_1 \longrightarrow P_2 \quad (4)$$

$$P \longrightarrow P' \Rightarrow \nu x P \longrightarrow \nu x P' \quad (5)$$

External vs internal mobility

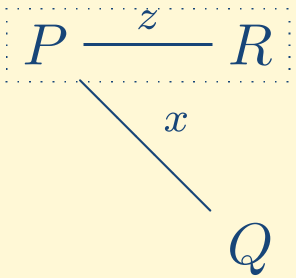
External mobility: comunication of free names

- Example: $P = \overline{c}\langle a \rangle .P'$

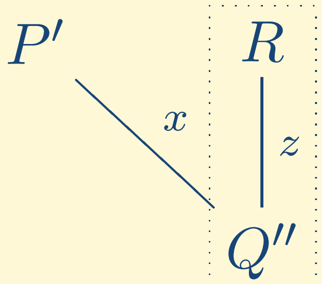
Internal mobility: comunication of private (fresh) names

- Example: $P = \nu z(\overline{a}\langle z \rangle .P')$
- the source of the expressive power of the π-calculus [Sangiorgi95]

Example



If $\nu z(P|R)|Q$ and if $P = \overline{x}\langle z\rangle.P'$ with z non-free in P' e $Q = x(y).Q'$ then



$P'|\nu z(Q''|R)$ with $Q'' = Q'\{z/y\}$

Actions - notation

$\alpha ::= \quad \overline{u}\langle y \rangle$ free output $\mid xy$ input

$\quad\quad\; \mid \nu z \overline{u}\langle z \rangle$ bound output $\mid \tau$ internal

we write $\overline{u}(z) =_{def} \nu z \overline{u}\langle z \rangle$

α	$subj(\alpha)$	$obj(\alpha)$	$fn(\alpha)$	$bn(\alpha)$	$n(\alpha)$
$\overline{x}\langle y \rangle$	x	y	$\{x, y\}$	$\emptyset$	$\{x, y\}$
xy	x	y	$\{x, y\}$	$\emptyset$	$\{x, y\}$
$\overline{x}(z)$.	x	z	$\{x\}$	$\{z\}$	$\{x, z\}$
τ	$-$	$-$	$\emptyset$	$\emptyset$	$\emptyset$

Labelled Transition System - 1

$$\frac{}{u(x).P \xrightarrow{uy} P\{y/x\}}\ in \quad \frac{}{\overline{u}\langle x\rangle.P \xrightarrow{\overline{u}\langle x\rangle} P}\ out \quad \frac{}{\tau.P \xrightarrow{\tau} P}\ tau$$

$$\frac{P \xrightarrow{\alpha} P' \quad bn(\alpha) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\alpha} P'|Q}\ p-l \quad \frac{P \xrightarrow{\overline{x}\langle y\rangle} P' \quad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'}\ c-l$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'[sum-l]}$$

Labelled Transition System - 2

$$\frac{P \xrightarrow{\overline{x}(z)} P' \quad Q \xrightarrow{xz} Q'}{P|Q \xrightarrow{\tau} \nu z(P'|Q')} \, cl-l \qquad \frac{P \xrightarrow{\overline{x}\langle z\rangle} P'}{\nu z P \xrightarrow{\overline{x}(z)} P'} \, open$$

$$\frac{P \xrightarrow{\alpha} P' \quad z \notin n(\alpha)}{\nu z P \xrightarrow{\alpha} \nu z P'} \, res \qquad \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'|!P} \, r-act$$

$$\frac{P \xrightarrow{\overline{x}\langle y\rangle} P' \quad P \xrightarrow{xy} P''}{!P \xrightarrow{\tau} (P'|P'')|!P} \, r-c \qquad \frac{P \xrightarrow{\overline{x}(z)} P' \quad P \xrightarrow{xz} P'' \quad z \notin fn(P)}{!P \xrightarrow{\tau} (\nu z(P'|P''))|!P} \, r-cl$$

Labelled Transition System - 3

Harmony lemma:

- $P \equiv \xrightarrow{\alpha} P' \Rightarrow P \xrightarrow{\alpha} \equiv P'$
- $P \longrightarrow P' \iff P \xrightarrow{\tau} \equiv P'$

Strong bisimilarity: Strong bisimilarity is the largest binary symmetric relation $\sim$ on processes such that whenever $P \sim Q$, if $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \sim Q'$.

Weak transition relations

- $\Longrightarrow$ is the reflexive and transitive closure of $\xrightarrow{\tau}$
- $\stackrel{\alpha}{\Longrightarrow}$ is $\Longrightarrow \xrightarrow{\alpha} \Longrightarrow$, where α is an action.

Weak bisimilarity: weak bisimilarity is the largest binary symmetric relation $\approx$ on processes such that whenever $P \approx Q$, if $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \approx Q'$

Properties:

- $\sim \subseteq \approx$
- $\approx$ is an equivalence relation

BUT

They are not contextual congruence relations

- let $P = \overline{x}|y$ and $Q = \overline{x}.y + y.\overline{x}$ be two processes
- let $C[\,] = a(x).[\,]$ be a context

we have

- $P \sim Q$ but $C[P] \not\sim C[Q]$

Discussion

π-calculus

- it models concurrent mobile computation
- it has an intuitive notion of obsevational equivalence (bisimilarity)

But

- reduction is in general not confluent
- there is no canonical normal form
- bisimilarity is not a contextual equivalence relation

Solutions:

- restricting the calculus (this talk)
- extending the calculus
  - Fusion calculus (Parrow Victor)
  - Solos (Laneve Parrow Victor)

From π-calculus to linear π-calculus

We add the following restrictions:

Asynchrony: the act of sending a datum and the act of receiving it are separate. In particular no process need to wait to send a datum.

Locality: channels received in input are only used for output and dually channels sent in output are only used for input

Internal mobility: the communication involves only private (fresh) names

Linearity: The linearity constraint gives a discipline on how much a process can use a name.

Syntax:

$$P ::= a(\vec{x}).P \mid \overline{a}\langle\vec{t}\rangle \mid P|P \mid \nu\vec{x}P \mid !P|0$$

Reduction:

$$a(\vec{x}).P|\overline{a}\langle\vec{t}\rangle \longrightarrow P\{\vec{t}/\vec{x}\}$$

Output is not blocking and can be sequentialized:

$$\nu y, z(\overline{x}\langle y\rangle|\overline{y}\langle z\rangle|\overline{z}\langle a\rangle|R)$$

with $y, z \notin fn(R)$

Expressing synchrony in $A\pi$-calculus

The translation $[\![\cdot]\!]$ from π to $A\pi$ is an homomorphism except that

$$[\![\overline{a}\langle\vec{t}\rangle.P]\!] = \nu y(\overline{a}\langle\vec{t}, y\rangle | y().[\![P]\!])$$
$$[\![a(\vec{x}).P]\!] = a(\vec{x}, y).(\overline{y}\langle\,\rangle | [\![Q]\!])$$

The trasition $\overline{a}\langle\vec{t}\rangle.P|a(\vec{x}).Q \xrightarrow{\tau} P|Q\{\vec{t}/\vec{x}\}$ is mimicked by the sequence

$$\begin{array}{rcl} [\![\overline{a}\langle\vec{t}\rangle.P|a(\vec{x}).Q]\!] & \equiv & \nu y((\overline{a}\langle\vec{t}, y\rangle | y().[\![P]\!])|a(\vec{x}, z).(\overline{z}\langle\,\rangle | [\![Q]\!])) \\ & \longrightarrow & \nu y(y().[\![P]\!]|(\overline{z}\langle\,\rangle | [\![Q]\!])\{\vec{t}/\vec{x}, y/z\}) \\ & \equiv & \nu y(y().[\![P]\!]|(\overline{y}\langle\,\rangle | [\![Q]\!])\{\vec{t}/\vec{x}\}) \\ & \longrightarrow & [\![P]\!]|[\![Q\{\vec{t}/\vec{x}\}]\!] \end{array}$$

Syntax:

$$P ::= a(\vec{x}).P \,|\, \overline{a}\langle\vec{t}\rangle \,|\, P|P \,|\nu\vec{x}P \,|!P|0$$

where we impose that in the process $a(\vec{x}).P$ the names $\vec{x}$ cannot occur free in P as a subject of an input prefix. (**Output capability constraint**)

Locality: Every process that receive via a name is local to the process that create that name.

Example: a printer manager process can comunicate to another process the capability to send a job to the printer but not the capability to receive a job for the printer

Expressiveness : the same as $A\pi$ ([Boreale98])

Only internal mobility is allowed

Syntax: we pose $\overline{a}(\vec{x}).P =_{def} \nu\vec{x}(\overline{a}\langle\vec{x}\rangle.P)$

$P ::=$	$a(\vec{x}).P$	input
	$\overline{a}(\vec{x}).P$	output
	$P\|P$	parallel composition
	$\nu\vec{x}\,P$	restriction
	$K\langle\vec{x}\rangle$	recursion
	0	termination

where we impose that

- in $\overline{a}(\vec{x}).P$ the names $\vec{x}$ can occur in P only as the subject of *input* prefixes.
- in $a(\vec{x}).P$ the names $\vec{x}$ can occur in P only as the subject of an *output*.

Reduction: $a(\vec{x}).P|\overline{a}(\vec{x}).Q \longrightarrow \nu\vec{x}P|Q$

Expressivness: The same as $LA\pi$-calculus ([Boreale98])

Linear π-calculus [YoshidaHondaBerger03]

Syntax: we pose $\overline{a}(\vec{x})\, P =_{def} \nu\vec{x}(\overline{a}\langle\vec{x}\rangle | P)$

$P ::=$	$a(\vec{x}).P$	input
	$\overline{a}(\vec{x})\, P$	asynchronous output
	$!a(\vec{x}).P$	guarded replication
	$\ldots$	and the same as above

structural and reduction rules:

$$\overline{x}(\vec{u})\,\overline{y}(\vec{v})\, P \equiv \overline{y}(\vec{v})\,\overline{x}(\vec{u})\, P \text{ se } x \notin \vec{v} \text{ e } y \notin \vec{u}$$

$$\overline{x}(\vec{u})\,(P|Q) \equiv (\overline{x}(\vec{u})\, P)|Q \text{ se } \vec{u} \notin Q$$

$$\nu y\, \overline{x}(\vec{u})\, P \equiv \overline{x}(\vec{u})\, \nu y\, P \text{ se } y \notin \{x, \vec{u}\}$$

$$a(\vec{x}).P|\overline{a}(\vec{x})\, Q \longrightarrow \nu\vec{x}(P|Q)$$

$$!a(\vec{x}).P|\overline{a}(\vec{x})\, Q \longrightarrow !a(\vec{x}).P|\nu\vec{x}(P|Q)$$

$$P \longrightarrow P' \Rightarrow \overline{a}(\vec{x})\, P \longrightarrow \overline{a}(\vec{x})\, P'$$

Action and channel modes

- input modes
  - $\downarrow$ linear (ex. $a(x).P$)
  - $!$ replicated (ex. $!a(x).P$)
- output modes (ex $\overline{a}(x)\ P$)
  - $\uparrow$ linear
  - $?$ replicated
- neutral ($\updownarrow$)

We use pO (resp. pI) to vary on output (resp. input) modes

Types and type environments

$$\tau \ ::= \ \tau_I | \tau_O | \updownarrow \quad \tau_I ::= (\vec{\tau_O})^{pI} \quad \tau_O ::= (\vec{\tau_I})^{pO}$$
$$\Gamma \ ::= \ a_1 : \tau_1, \ldots, a_n : \tau_n$$

- Γ is always well formed (each name appear only once)
- $Dom(\Gamma)$ is the set of names that occur in Γ. If $\Gamma = \Delta, a : \tau$ we note $\Gamma(a) = \tau$
- Given a type τ, we define its **dual** $\tau^\perp$ in the following way
  - $((\vec{\tau_I})^\downarrow)^\perp = (\vec{\tau_I^\perp})^\uparrow$
  - $((\vec{\tau_I})^?)^\perp = (\vec{\tau_I^\perp})^!$
  - $((\vec{\tau_O})^\uparrow)^\perp = (\vec{\tau_O^\perp})^\downarrow$
  - $((\vec{\tau_O})^!)^\perp = (\vec{\tau_O^\perp})^?$

Composition of type environments

Let Γ and Δ two type environments such that, for all $a \in Dom(\Gamma) \cap Dom(\Delta)$ we have that one of the following holds

- $\Gamma(a) = \Delta(a)^{\perp}$
- $\Gamma(a) = \Delta(a) = (\vec{\tau_I})^?$

Then we define $\Gamma \odot \Delta$ the environment Ξ such that $Dom(\Xi) = Dom(\Gamma) \cup Dom(\Delta)$ and

$$\Xi(a) = \begin{cases} \Gamma(a) & \text{if } a \in Dom(\Gamma) \setminus Dom(\Delta) \\ \Delta(a) & \text{if } a \in Dom(\Delta) \setminus Dom(\Gamma) \\ \updownarrow & \text{if } \Gamma(a) = \Delta(a)^{\perp} \text{ with a linear modality} \\ (\tau_O)^! & \text{if } \Gamma(a) = \Delta(a)^{\perp} \text{ with a replicated modality} \\ (\tau_I)^? & \text{if } \Gamma(a) = \Delta(a) = (\tau_I)^? \end{cases}$$

Typing rules

$$\frac{}{0\triangleright}\ zero \qquad \frac{P_1 \triangleright \Gamma_1 \quad P_2 \triangleright \Gamma_2}{P_1|P_2 \triangleright \Gamma_1 \odot \Gamma_2}\ par \qquad \frac{P \triangleright \Gamma, x : \tau \quad md(\tau) \in \{\updownarrow, !\}}{\nu x P \triangleright \Gamma}\ res$$

$$\frac{P \triangleright \Gamma \quad md(\tau) \in \{\updownarrow, ?\}}{P \triangleright \Gamma, x : \tau}\ weak \qquad \frac{P \triangleright \vec{x} : \vec{\tau_O}, \downarrow \Gamma_1, ?\Gamma_2}{a(\vec{x}).P \triangleright \uparrow \Gamma_1, ?\Gamma_2, a : (\vec{\tau_O})^{\downarrow}}\ in^{\downarrow}$$

$$\frac{P \triangleright \vec{x} : \vec{\tau_O}, ?\Gamma}{!a(\vec{x}).P \triangleright ?\Gamma, a : (\vec{\tau_O})^{!}}\ in^{!} \qquad \frac{P \triangleright \Gamma, \vec{x} : \vec{\tau_I}}{\overline{a}(\vec{x})\ P \triangleright \Gamma \odot (a : (\vec{\tau_I})^{pO})}\ out$$

Adding constraints

The typing system above guarantees the confluence property ($\longrightarrow$ is Church-Rosser).
We can add constraints:

acyclicity: strong normalization [YoshidaHondaBerger03]
sequentiality: full abstraction in PCF [YoshidaHondaBerger01]

We can also extend the calculus with:

branching/selection: additives
non determinism: see [VaraccaYoshida06]