The Polyadic π - calculus : a Tutorial

Robin Milner

Computer Science Department,
University of Edinburgh, UK

October 1991

A model for Concurrent Computation

A deterministic sequential process modelled by

- Functional View
- λ-calculus (arguments $->$ result)

None of them can be used to model concurrent computation mainly since non-determinism is inherent in a concurrent system

π-Calculus

Objective : To describe and analyze systems consisting of agents which interact among themselves and whose configuration is continually changing.

Based on the notion of *naming*

- Naming strongly presupposes independence
- One naturally assumes that the *namer* and *named* are co-existing entities.
- Act of using a name, or address is inextricably confused with the act of communication.

Basic Idea

There are two types of entities :

- *name:*

  The most primitive entity.

  Names, infinitely many, are $x, y, ..., \in \mathcal{X}$.

  They have no structure.

- *Process:*

  Built from names by the syntax :

$$P ::= \sum_{i \in I} \pi_i . P_i \quad | \quad P|Q \quad | \quad !P \quad | \quad (\nu x)P$$

$\underline{\sum_{i\in I} \pi_i.P_i}$

The prefix π represents an atomic action.

Two basic forms of prefix:

- $x(y)$: "input some name -call it y -along link named x"
- $\overline{x}y$: "output the name y along the link named x.

A channel is named and not a process

$P|Q$: analogous to "$|$" operator in CCS

- P and Q are concurrently active.
- They can act independently.
- They can also communicate.

$\underline{!P}$

- The "bang" P - means $P \mid P \mid P \cdots$; as many copy as we wish.
- The operator ! is called *replication.* A common instance of replication :

  $!\pi P$ - a resource which can only be replicated when a requester communicates via π.

$(\nu x)P$: analogous to restriction operator in CCS

- restricts the use of the name x to P.
- it declares a new unique name x, distinct from all external names, for use in P.

Examples

$$\overline{x}y.0 \mid x(v).\overline{u}v.0 \mid \overline{x}z.0$$

$$(\nu x)(\overline{x}y \mid x(u).\overline{u}v) \mid \overline{x}z$$

$$\overline{x}y \mid !x(u).\overline{u}v \mid \overline{x}z$$

Free names and bound names

There are two binding operators the input prefix, and the restriction. *bound* and *free* names for a prefix are :

- For the input prefix : $x(y)$ which binds y.

$$\mathrm{bn}(x(y)) = \{y\}, \qquad \mathrm{fn}(x(y)) = \{x\}$$

- For the output prefix : $\overline{x}y$

$$\mathrm{bn}(\overline{x}y) =, \qquad \mathrm{fn}(\overline{x}y) = \{x, y\}$$

Also the *names* associated with a process are

$$\mathrm{n}(P) \stackrel{\mathrm{def}}{=} \mathrm{bn}(P) \cup \mathrm{fn}(P)$$

Structural Congruence

The smallest congruence relation, $\equiv$ such that the following laws hold:

- $(\mathcal{N}/\equiv, *, 0)$ is a commutative monoid for $* \in \{+, |\}$.
  (*like in CCS*)
- $!P \equiv P \,|!P$
- $(\nu x)0 \equiv 0, (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$
- if $x \notin \mathrm{fn}(P)$ then $(\nu x)(P \mid Q) \equiv P \mid (\nu x)Q$

Let us show that $(\nu x P) \equiv P$ when $x \notin \mathrm{fn}(P)$.

Reduction Rules

Reduction relation $\rightarrow$ over processes($P \rightarrow P'$).

- *Communication* (the only axiom) :

$$(\cdots + x(y).P) \mid (\cdots + \overline{x}z.Q) \rightarrow P\{z/y\} \mid Q$$

- *Parallel composition* law : *Restriction* law :

$$\frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \qquad \frac{P \rightarrow P'}{(\nu x)P \rightarrow (\nu x)P'}$$

- *Structural Congruence* law :

$$\frac{Q \equiv P, P \rightarrow P', P' \equiv Q'}{Q \rightarrow Q'}$$

Applications

- **Multiple inputs and outputs along same channel :**

$$x(yz) \mid \overline{x}y_1z_1 \mid \overline{x}y_2z_2$$

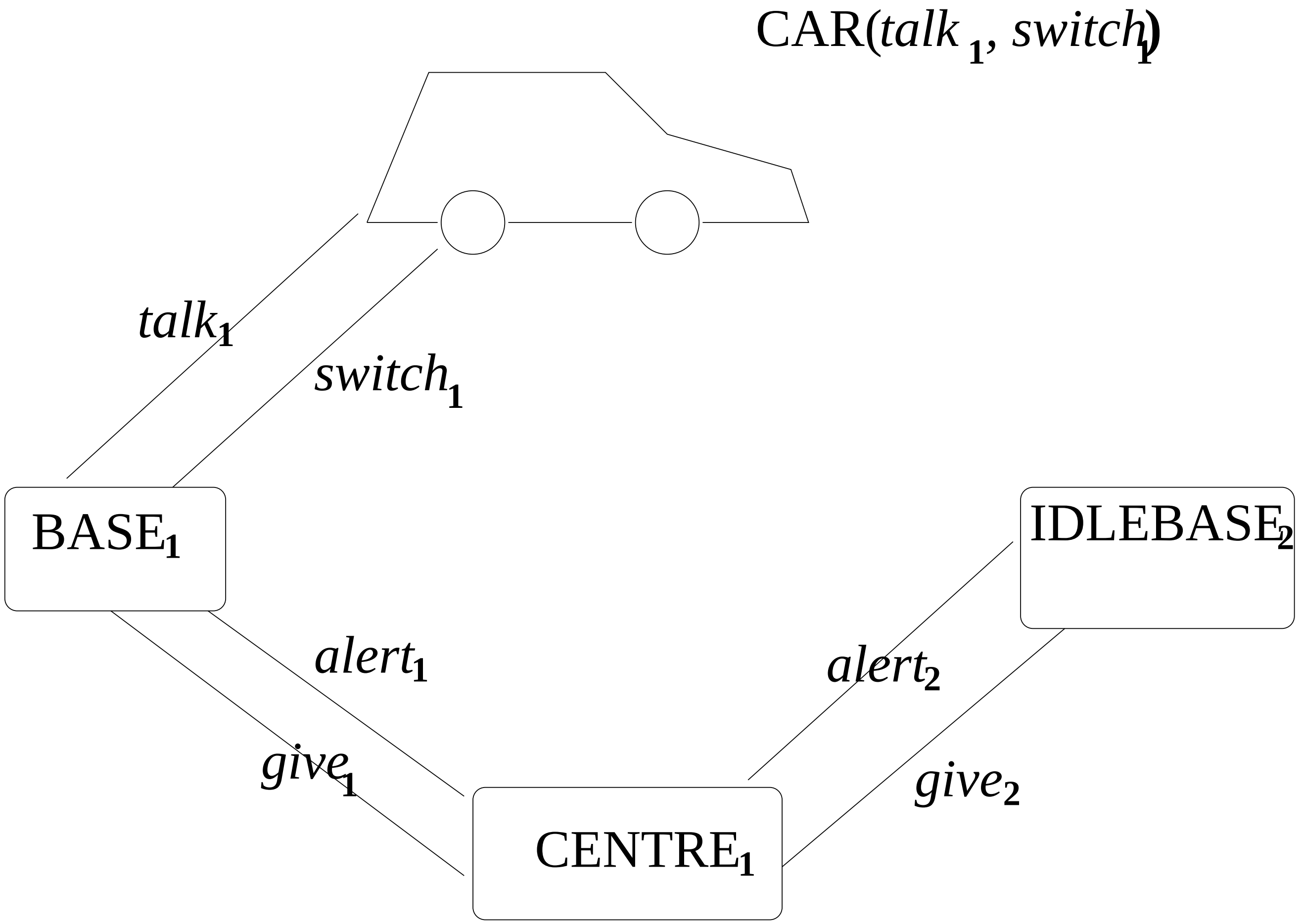
Intention : y, z get bound to either y_1, z_1 or y_1, z_2.
But there can be mix-up also.

Solution : Introduce abbreviations :

$$x(y_1 \cdots y_n) \quad \text{for} \quad x(w).w(y_1).\cdots.w(y_n)$$

$$\overline{x}y_1 \cdots y_n \quad \text{for} \quad (\nu w)\overline{x}w.\overline{w}y_1.\cdots.\overline{w}y_n$$

Mobile Telephones



The system is :

$$\mathrm{SYSTEM}_1 \stackrel{\mathrm{def}}{=} (\nu t_i, s_i, g_i, a_i : i = 1, 2)$$

$$(\mathrm{CAR}(t_1, s_1) \mid \mathrm{BASE}_1 \mid \mathrm{IDLEBASE}_2 | \mathrm{CENTER}_1)$$

The components are :

$$\begin{aligned} \mathrm{CAR}(talk, switch) \quad &\stackrel{\mathrm{def}}{=} \quad talk.\mathrm{CAR}(talk, switch) + \\ & \quad\quad switch(talk', switch').\mathrm{CAR}(talk', switch') \end{aligned}$$

$$\mathrm{BASE}_i \stackrel{\mathrm{def}}{=} talk_i.\mathrm{BASE}_i + give_i(t' s').\overline{switch_i} t' s'.\mathrm{IDLEBASE}_i$$

$$\mathrm{IDLEBASE}_i \stackrel{\mathrm{def}}{=} alert_i.\mathrm{BASE}_i$$

For the center ,

$$\mathrm{CENTRE}_1 \stackrel{\mathrm{def}}{=} \overline{give}_1 talk_2 switch_2.alert_2.\mathrm{CENTRE}_2$$

$$\mathrm{CENTRE}_2 \stackrel{\mathrm{def}}{=} \overline{give}_2 talk_1 switch_1.alert_1.\mathrm{CENTRE}_1$$

Transition : System1 $\rightarrow$ System2

$$
\begin{aligned}
\mathrm{S}_1 \quad &\equiv \quad (\nu c)\,(\mathrm{CAR}(talk_1, switch_1) \mid \mathrm{BASE}_1 \mid \mathrm{IDLEBASE}_2 \mid \mathrm{CENTRE}_1) \\
&\rightarrow \quad (\nu c)\,\big(\mathrm{CAR}(talk_1, switch_1) \mid \overline{switch_1}\, talk_2\, switch_2.\mathrm{IDLEBASE}_1 \mid \\
&\qquad \mathrm{IDLEBASE}_2 \mid alert_2.\mathrm{CENTRE}_2) \\
&\rightarrow \quad (\nu c)\,(\mathrm{CAR}(talk_2, switch_2) \mid \mathrm{IDLEBASE}_1 \mid \mathrm{IDLEBASE}_2 \mid \\
&\qquad alert_2.\mathrm{CENTRE}_2) \\
&\rightarrow \quad (\nu c)\,(\mathrm{CAR}(talk_2, switch_2) \mid \mathrm{IDLEBASE}_1 \mid \mathrm{BASE}_2 \mid \mathrm{CENTRE}_2) \\
&\equiv \quad \mathrm{S}_2
\end{aligned}
$$

Reduction equivalence and congruence

Definition : An agent B occurs *guarded* in A if it has some occurrence in A which is not under a prefix α. A process P is *observable* at α, written $P \downarrow_\alpha$, if some $\alpha.A$ occurs unguarded in P with α unrestricted.

Definition : *Strong Reduction Equivalence*, $\dot{\sim}_r$, is the largest equivalence relation $\mathcal{R}$ over processes such that $P \mathcal{R} Q$ implies :
1. If $P \rightarrow P'$, then $Q \rightarrow Q'$ for some Q' such that $P' \mathcal{R} Q'$.
2.For each α, if $P \downarrow_\alpha$ then $Q \downarrow_\alpha$.

Definition : *Strong reduction congruence* , $\sim_r$, is the largest congruence included in reduction equivalence.

Definition : *Weak Reduction Equivalence*, $\dot{\approx}_r$, is the largest equivalence relation $\mathcal{R}$ over processes such that $P\mathcal{R}Q$ implies :
1. If $P \rightarrow P'$, then $Q \rightarrow^* Q'$ for some Q' such that $P'\mathcal{R}Q'$.
2.For each α, if $P \downarrow_\alpha$ then $Q \rightarrow^* \downarrow_\alpha$.

Commitment and Congruence

We shall call $\alpha.A$ a *commitment*; it is a process committed to act at α.

Definition : The *commitment* relation $\succ$ between processes and commitments is the smallest relation satisfying the following rules :

$$\text{SUM : } \cdots + \alpha.A \succ \alpha.A$$

$$\text{COMM : } \frac{P \succ x.F \qquad Q \succ \overline{x}.C}{P \mid Q \succ \tau.(F|C)}$$

$$\text{PAR : } \frac{P \succ \alpha.A}{P \mid Q \succ \alpha.(A \mid Q)} \qquad \text{RES : } \frac{P \succ \alpha.A}{(\nu x)P \succ \alpha.(\nu x)A} (\alpha \notin \{x, \overline{x}\})$$

$$\text{STRUCT : } \frac{Q \equiv P, P \succ \alpha.A, P' \equiv Q'}{Q \succ \alpha.B}$$

Definition $\dot\sim$: A relation $\mathcal{R}$ over agents is (strong) *simulation* if it includes $\equiv$, and also if $P\mathcal{R}Q$ and $P \succ \alpha.A$, then $Q \succ \alpha.B$ for some B such that $A\mathcal{R}B$.

Definition : $\mathcal{R}$ is a strong bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are strong simulations.
Note that $\dot\sim$ is not preserved under substitution. For example

$$\overline{x} \mid y \dot\sim \overline{x}.y + y.\overline{x} \qquad \text{But}$$

$$\overline{x} \mid x \not\dot\sim \overline{x}.x + x.\overline{x}$$

Definition : P and Q are strongly congruent, written $P \sim Q$, if $P\sigma \dot\sim Q\sigma$ for all substitutions σ.

Properties of Replication(the “bang” operator)

Proposition : $!P \mid !P \sim !P$.

Proof : There is a bisimulation whose process part consists of all pairs

$$(\nu y)(!P \mid !P \mid Q), (\nu y)(!P \mid Q)$$

for any y, P and Q.

- Take y to be empty.
- $Q \equiv 0$.

This ensures $!P \mid !P \dot{\sim} !P$. Moreover, it is a congruence.

Is it true that :

$$(\nu x)(P_1 \mid P_2 \mid !x.F) \sim (\nu x)(P_1 \mid !x.F) \mid (\nu x)(P_2 \mid !x.F)?$$

No !! The contradiction is :

$$\text{Take} \quad P_1 \equiv \overline{x}.y \quad \text{and} \quad P_2 \equiv x.0$$

P_1 can access the resource as well as interact with P_2 on the left-hand side. But on the right side P_1 can only access the resource.

Proposition : Assume that every free occurrence of x in P_1, P_2 and F is as a negative subject. Then

$$(\nu x)(P_1 \mid P_2 \mid !x.F) \sim (\nu x)(P_1 \mid !x.F) \mid (\nu x)(P_2 \mid !x.F)$$

Proof : There is a bisimulation consisting of pairs :

$$(\nu x)\,(P_1 \mid P_2 \mid !x.F)\,,(\nu x)\,(P_1 | !x.F) \mid (\nu x)\,(P_2 \mid !x.F)$$

for any P_1, P_2, x, F such that x occurs free in P_1, P_2 and F only as a negative subject.

Axiomatization of $\dot{\sim}$

For agents without replication,

Definition The theory SGE (Strong Ground Equivalence) is the smallest set of equations $A = B$ over agents satisfying the following :

- If $A_1 \equiv A_2$ then SGE $\quad A_1 = A_2$.
- SGE is closed under summation :
  SGE $\quad M_1 = M_2$, then SGE $\quad M_1 + N = M_2 + N$
- If SGE $\quad Fy = Gy$ for every y then SGE $\quad F = G$.
- SGE $\quad M + M = M$

- SGE $(\nu x) \sum \alpha_i \cdot A_i = \sum \alpha_i \cdot (\nu x) A_i$, if no α_i is either x or $\overline{x}$.
- (Expansion :)

$$
\begin{array}{llll}
\text{SGE} & M \mid N & = & \sum\{\alpha \cdot (A|N) : \alpha \cdot A \text{is a summand of} M\} \\
 & & + & \sum\{\beta \cdot (B \mid M)) : \beta.B \text{ is a summand of} N\} \\
 & & + & \sum\{\tau.(F \cdot C) : x.F \text{is a summand of } M \\
 & & & \quad \text{and } \overline{x} \text{ is a summand of N}\}
\end{array}
$$