Course on

Probabilistic Methods in Concurrency

(Concurrent Languages for Probabilistic Asynchronous Communication)

Lecture 1

The pi-calculus and the asynchronous pi-calculus.

Catuscia Palamidessi

INRIA Futurs & LIX

France

catuscia@lix.polytechnique.fr

Administrativia

- Homepage of the course:
  www.lix.polytechnique.fr/~catuscia/teaching/Pisa/
  - Slides
  - Some copies of the papers/books used as references

- Exam

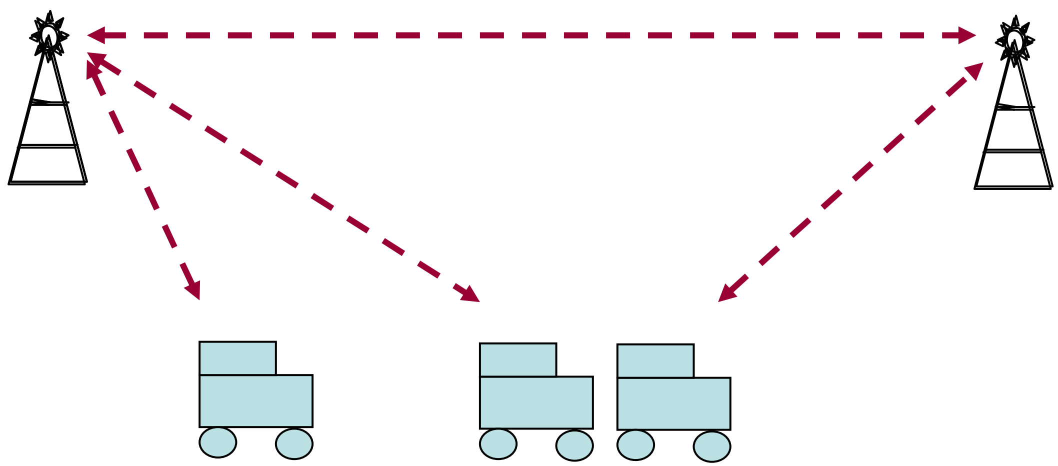
- Schedule

Plan of the lectures

1. The pi-calculus and the asynchonous pi-calculus
2. The pi-calculus hierarchy: encodings
   - Encoding of output prefix in the asynchonous pi-calculus
   - Encoding of input guarded choice in the asynchonous pi-calculus
3. The pi-calculus hierarchy: separation results
   - Separation between the pi-calculus and the asynchonous pi-calculus
   - Separation between the pi-calculus and CCS
4. Problems in distributed algorithms for which only randomized solutions exists
5. Basics of Measure Theory and Probability Theory
6. Probabilistic Automata
7. The probabilistic pi-calculus
8. Encoding of the pi-calculus into the asynchronous pi-calculus
9. Other uses of randomization: randomized protocols for anonymity and contract signing.
10. A proof search specification of the pi-calculus (speaker: Dale Miller)

The π-calculus

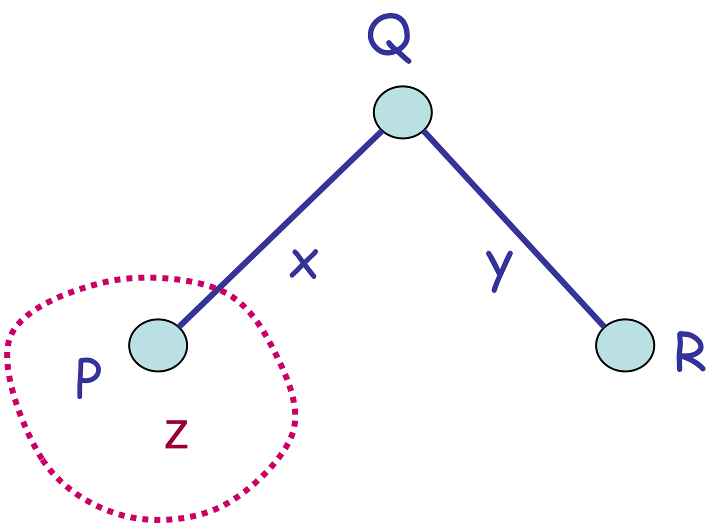
- Milner, Parrow, Walker 1989
- A concurrent calculus where the communication structure among existing processes can change over time.
  - Link mobility.



The π calculus: scope extrusion

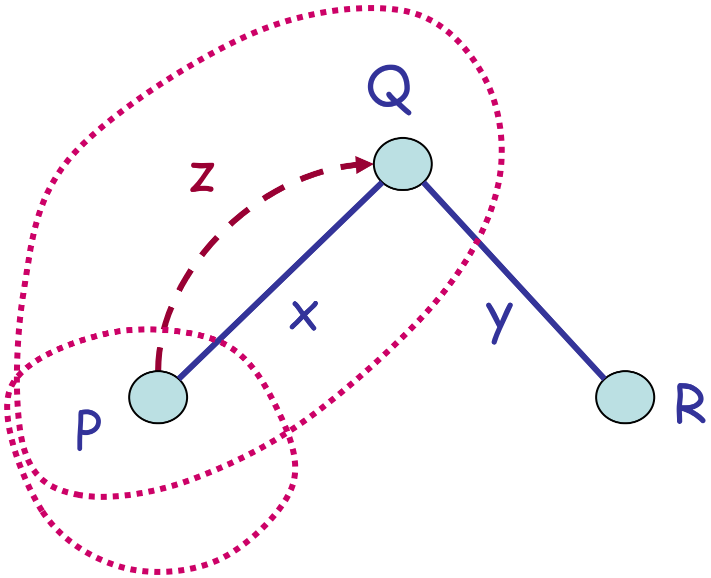
- A private channel name can be communicated and its scope can be extended to include the recipient
  - **Channel:** the name can be used to communicate
  - **Privacy:** no one else can interfere

- An example of link mobility:



The π calculus: scope extrusion

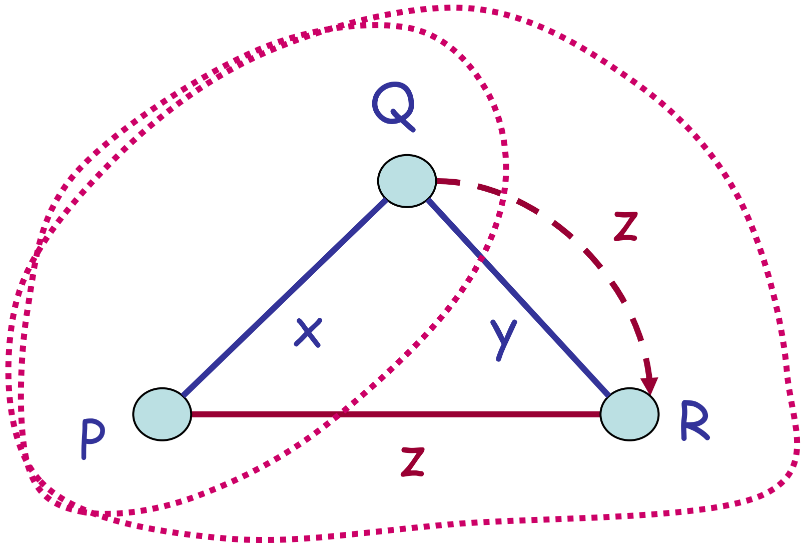
- A private channel name can be communicated and its scope can be extended to include the recipient
  - **Channel:** the name can be used to communicate
  - **Privacy:** no one else can interfere
- An example of link mobility:



The π calculus: scope extrusion

- A private channel name can be communicated and its scope can be extended to include the recipient
  - **Channel:** the name can be used to communicate
  - **Privacy:** no one else can interfere

- An example of link mobility:



The π calculus: some suggested bibliography

- Robin Milner. *Communicating and mobile systems: the pi-calculus.* Cambridge University Press, 1999

- Benjamin Pierce. Foundational Calculi for Programming Languages. Chapter in the *CRC Handbook of Computer Science and Engineering*, 1996

- Davide Sangiorgi and David Walker. *The pi-calculus. A Theory of Mobile Processes.* Cambridge University Press, 2001

- Joachim Parrow. An Introduction to the pi-Calculus. In Handbook of Process Algebra, ed. Bergstra, Ponse, Smolka, pages 479-543, Elsevier 2001. BRICS RS 99-42

The π-calculus: syntax

- Similar to CCS with value passing, but values are channel names, and recursion is replaced by replication (!)

$$\pi \ ::= \ x(y) \mid \bar{x}y \mid \tau$$

action prefixes (input, output, silent)
x, y are channel names

$$\begin{array}{llll} P & ::= & O & \text{inaction} \\ & \mid & \pi.P & \text{prefix} \\ & \mid & P \mid P & \text{parallel} \\ & \mid & P + P & \text{sum} \\ & \mid & (\nu x)P & \text{restriction, new name} \\ & \mid & !\,P & \text{replication} \end{array}$$

The π-calculus: syntax

- Names: $n(P)$
  - Free $fn(P)$
  - Bound $bn(P)$
    - Input and restriction are binders
  - Exercise: give the formal definition of $fn(P)$ and $bn(P)$

Example: $P = ((\nu x)\bar{y}x.x(z).\bar{z}x.0) \mid (y(w).\bar{w}u.0)$

we have: $fn(P) = \{y, u\}$, $bn(P) = \{x, z, w\}$

- Alpha conversion

Example: $Q = ((\nu v)\bar{y}v.v(z).\bar{z}v.0) \mid (y(x).\bar{x}u.0)$

we have: $P \equiv_\alpha Q$

The π-calculus: structural equivalence

- Introduced to simplify the description of the operational semantics
  - If $P \equiv_\alpha Q$ then $P \equiv Q$
  - $P \mid Q \equiv Q \mid P$
  - $P + Q \equiv Q + P$
  - $!\,P \equiv P \mid !\,P$

- Some presentations include other equivalences, for instance:
  - $P \mid 0 \equiv P$, $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$
  - $P + 0 \equiv P$, $(P + Q) + R \equiv P + (Q + R)$, $P + P \equiv P$
  - $(\nu\, x)\, (\nu\, y)\, P \equiv (\nu\, y)\, (\nu x)\, P$, $(\nu\, x)\, P \equiv P$ if $x \notin fn(P)$
  - $P \mid (\nu\, x)\, Q \equiv (\nu\, x)\, (P \mid Q)$ if $x \notin fn(P)$ (scope extrusion)

The π–calculus: operational semantics

- The operational semantics of the π-calculus is defined as a labeled transition system. Transitions have the form

$$P \xrightarrow{\mu} Q$$

  Here P and Q are processes and μ is an action

- There are various operational semantics for the π-calculus. We describe here the late semantics. Actions are defined as follows:

μ	kind	$fn(\mu)$	$bn(\mu)$
τ	silent	$\emptyset$	$\emptyset$
$x(y)$	(bound) input	$\{x\}$	$\{y\}$
$\bar{x}y$	free output	$\{x, y\}$	$\emptyset$
$\bar{x}(y)$	bound output	$\{x\}$	$\{y\}$

The π–calculus: late semantics

Cong $$\frac{P' \equiv P \qquad P \xrightarrow{\mu} Q \qquad Q \equiv Q'}{P' \xrightarrow{\mu} Q'}$$

Prefix $$\frac{}{\alpha.P \xrightarrow{\alpha} P}$$

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

Sum $$\frac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P'}$$

Res $$\frac{P \xrightarrow{\mu} P'}{\nu y P \xrightarrow{\mu} \nu y P'} \quad y \notin n(\mu)$$

Open $$\frac{P \xrightarrow{\bar{x}y} P'}{\nu y P \xrightarrow{\bar{x}(y)} P'} \quad x \neq y$$

L-Com $$\frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\bar{x}z} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'}$$

Close $$\frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\bar{x}(y)} Q'}{P|Q \xrightarrow{\tau} \nu y(P'|Q')}$$

Questions: 1) Why the side condition in Par?
2) Could we write x(z) in L-Com and avoid the substitution?

The π–calculus: early semantics

- New kind of action: free input xz
- Add E-input and replace L-Com by E-Com

Cong $$\frac{P' \equiv P \qquad P \xrightarrow{\mu} Q \qquad Q \equiv Q'}{P' \xrightarrow{\mu} Q'}$$

E-Input $$\frac{}{x(y).P \xrightarrow{xz} P\{z/y\}}$$

Prefix $$\frac{}{\alpha.P \xrightarrow{\alpha} P}$$

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

Sum $$\frac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P'}$$

Res $$\frac{P \xrightarrow{\mu} P'}{\nu y P \xrightarrow{\mu} \nu y P'} \quad y \notin n(\mu)$$

Open $$\frac{P \xrightarrow{\bar{x}y} P'}{\nu y P \xrightarrow{\bar{x}(y)} P'} \quad x \neq y$$

E-Com $$\frac{P \xrightarrow{xy} P' \qquad Q \xrightarrow{\bar{x}y} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Close $$\frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\bar{x}(y)} Q'}{P|Q \xrightarrow{\tau} \nu y(P'|Q')}$$

The π–calculus: late bisimulation

- **Definition** We say that a binary relation $\mathcal{S}$ is a late simulation if $P\,\mathcal{S}\,Q$ implies that

  1. if $P \xrightarrow{\mu} P'$ and μ is τ or output, with $bn(\mu) \cap fn(P,Q) = \emptyset$, then for some Q', $Q \xrightarrow{\mu} Q'$ and $P'\,\mathcal{S}\,Q'$.

  2. if $P \xrightarrow{x(y)} P'$ and $y \notin fn(P,Q) = \emptyset$, then for some Q', $Q \xrightarrow{x(y)} Q'$ and for all z, $P'\{z/y\}\,\mathcal{S}\,Q'\{z/y\}$.

- The relation $\mathcal{S}$ is a late bisimulation iff both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are late simulations.

- P and Q are late bisimilar, notation $P \sim_L Q$, iff $P\,\mathcal{S}\,Q$ for some late bisimulation $\mathcal{S}$.

The π–calculus: early bisimulation

Definition

- We say that a binary relation $\mathcal{S}$ is an early simulation if $P\,\mathcal{S}\,Q$ implies that

  if $P \xrightarrow{\mu} P'$ and μ is any action with $bn(\mu) \cap fn(P,Q) = \emptyset$, then for some Q', $Q \xrightarrow{\mu} Q'$ and $P'\,\mathcal{S}\,Q'$.

- The relation $\mathcal{S}$ is an early bisimulation iff both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are early simulations.

- P and Q are early bisimilar, notation $P \sim_E Q$, iff $P\,\mathcal{S}\,Q$ for some early bisimulation $\mathcal{S}$.

Late vs early bisimulation

Late bisimulation is strictly more discriminating than early bisimulation.

Example

$$P \equiv x(y).R + x(y).0$$

$$Q \equiv x(y).R + x(y).0 + x(y).\ \text{if } y = z \text{ then } R \text{ else } 0$$

We have that $P \sim_E Q$ but $P \not\sim_L Q$

Exercise: write a similar example without using the match operator (i.e. the if-then-else). Hint: use synchronization

Congruence

Question: are $\approx_L$, $\approx_E$ congruences?

Answer: **No.** Example:

$$x(z).0 \mid \bar{y}z.0 \sim_E x(z).\bar{y}z.0 + \bar{y}z.x(z).0$$

but

$$w(x)(x(z).0 \mid \bar{y}z.0) \not\sim_E w(x)(x(z).\bar{y}z.0 + \bar{y}z.x(z).0)$$

There are other equivalences which are defined to be congruences.
In particular Open bisimulation.
Cfr. lecture by Dale

The asynchronous π-calculus

- If P | Q is interpreted as the composition of two remote processes P and Q, then the mechanism of synchronous communication seems unrealistic

$$\bar{x}y.P \mid x(y).Q \xrightarrow{\tau} P \mid Q$$

- Synchronization combined with choice seems even less realistic

$$\begin{tikzcd} & P_1 \mid Q_2 \\ \bar{x}_1 y.P_1 + x_2(y).P_2 \mid \bar{x}_2 y.Q_1 + x_1(y).Q_2 \arrow[ur, "\tau"] \arrow[dr, "\tau"] & \\ & P_2 \mid Q_1 \end{tikzcd}$$

- In a distributed system, communication is asynchronous (exchange of messages). The send takes place independently of the readiness of a receiver, and it is not blocking

- The asynchnous π-calculus: A calculus for representing asynchnous communication. It was introduced independently by Honda-Tokoro [1991] and by Boudol [1992]

The asynchronous π-calculus: syntax

- **It differs from the π-calculus for the absence of the output prefix (replaced by output action) and also for the absence of the +**

$$\pi \ ::= \ x(y) \mid \tau$$

action prefixes (input, silent)

x, y are channel names

P	$::=$	O	inaction
	$\mid$	$\pi.P$	prefix
	$\mid$	$\bar{x}y$	output action
	$\mid$	$P \mid P$	parallel
	$\mid$	$(\nu x)P$	restriction, new name
	$\mid$	$!\,P$	replication

The asynchronous π-calculus: OS

- The operational semantics of the asynchronous π-calculus (π_a) are the same as those of the (synchronous) π-calculus (π), we only eliminate the rule for + and replace the output rule with the following:

$$Output \quad \frac{}{\bar{x}y \xrightarrow{\bar{x}y} 0}$$

- The early and late bisimulations are obtained as usual

- The interpretation is as follows:
  - The send takes place when the output action is at the top-level $(\nu y)(\bar{x}y \mid P) \mid Q$
  - The receive takes place when the output action matches a corresponding input, i.e. when we apply the rule comm or close

$$\bar{x}y \mid P \mid x(z).Q \longrightarrow^* \bar{x}y \mid P' \mid x(z).Q \longrightarrow P' \mid Q[z/y]$$