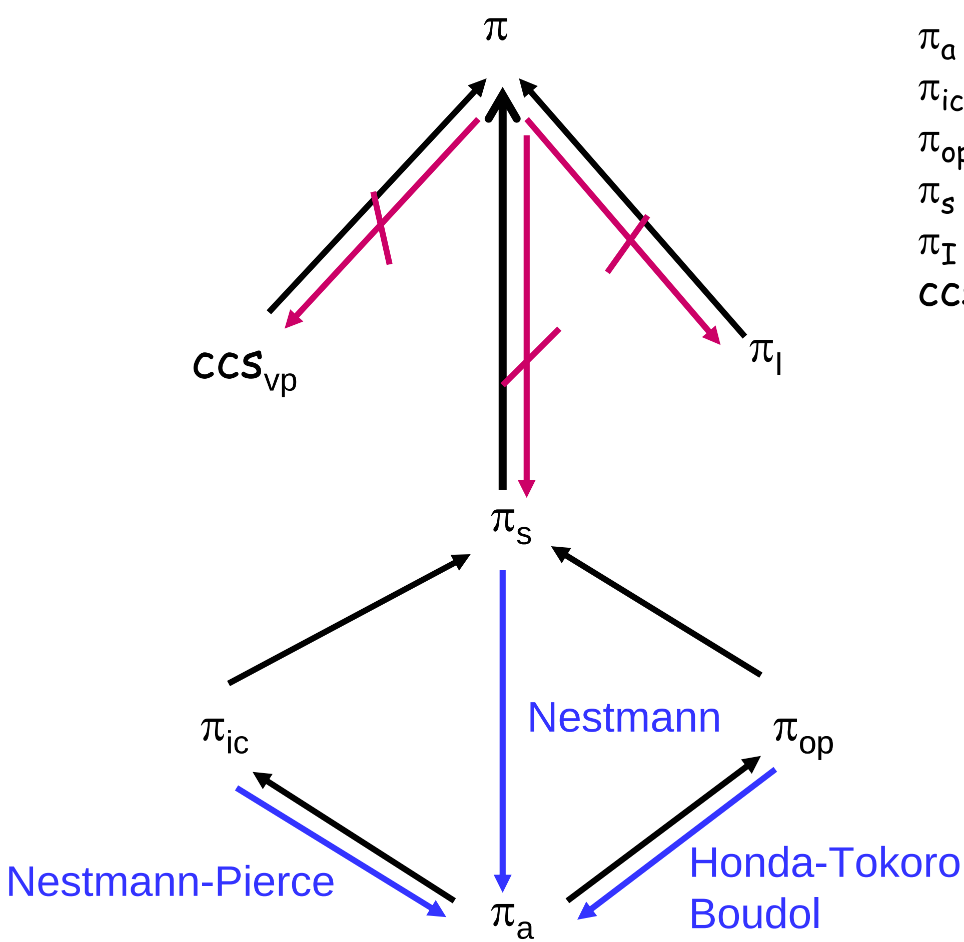
Probabilistic Methods in Concurrency

Lecture 3
The pi-calculus hierarchy: separation results

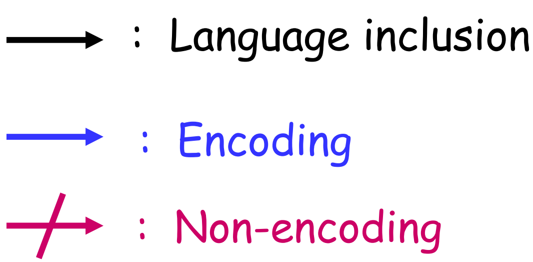
Catuscia Palamidessi
catuscia@lix.polytechnique.fr
www.lix.polytechnique.fr/~catuscia

Page of the course:
www.lix.polytechnique.fr/~catuscia/teaching/Pisa/

The π-calculus hierarchy



π_a : asynchronous π
π_{ic} : asynchronous π + input-guarded choice
π_{op} : asynchronous π + output prefix
π_s : asynchronous π + separate choice
π_I : π with internal choice (Sangiorgi)
ccs_{vp} : value-passing ccs



The separation between π and π_s

This separation result is based on the fact that it is not possible to solve the symmetric leader election problem in π_s, while it is possible in π

- Some definitions:
  - **Leader Election Problem (LEP):** All the nodes of a distributed system must agree on who is the leader. This means that in every possible computation, all the nodes must eventually output the name of the leader on a special channel *out*
    - No deadlock
    - No livelock
    - No conflict (only one leader must be elected, every process outputs its name and only its name)
  - **Symmetric LEP:** the LEP on a symmetric network
    - Hypergraphs and hypergraph associated to a network
    - Hypergraph automorphism
    - Orbits, well-balanced automorphism
    - Examples
    - Symmetry

The separation between π and π_s

- **Theorem:** If a network with at last two nodes has an automorphism $\sigma \neq \mathrm{id}$ with only one orbit, then it is not possible to write in π_s a symmetric solution to the LEP

- **Corollary:** The same holds if the authomorphism is well-balanced

- **Proof** (sketch). We prove that in π_s every system trying to solve the electoral problem has at least one diverging computation

  1. If the system is symmetric, then the first action cannot be $\overline{out}\, k$

  2. As soon as a process perform an action, let all the other processes in the same orbit perform the same action as well. At the end of the round in the orbit, the system is again symmetric.

Note that the system can change communication structure dynamically

The separation between π and π_s

- Crucial point: if the action performed by P_i is a communication with P_j in the same orbit, we need to ensure that P_j can do the same action afterwards.

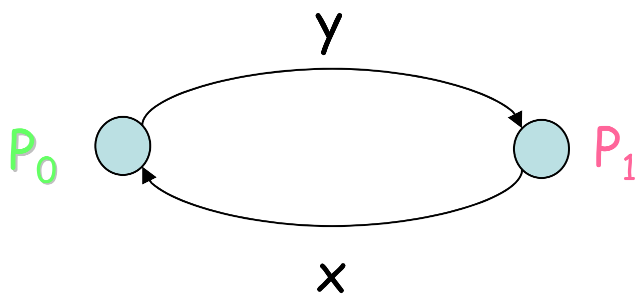
- This property holds in fact, due to the following:

- **Lemma:** Diamond lemma for π_s

- Note that in π (in π with mixed choice) the diamond lemma does not hold

The separation between π and π_s

- **Remark:** In π (in π with mixed choice) we can easily write a symmetric solution for the LEP in a network of two nodes:



$$P_0 = x.\overline{out}\,0 + \bar{y}.\overline{out}\,1$$

$$P_1 = y.\overline{out}\,1 + \bar{x}.\overline{out}\,0$$

The separation between π and π_s

- **Corollary:** there does not exists an encoding of π (π with mixed choice) in π_s which is homomorphic wrt | and renaming, and preserves the observables on every computation.

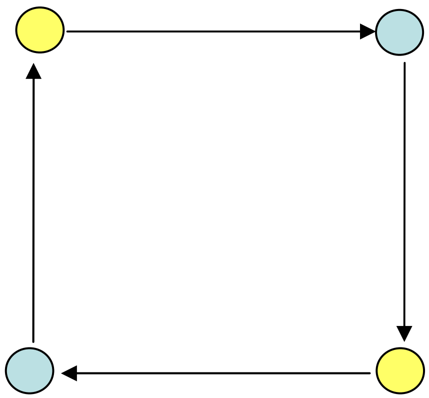
- **Proof** (scketch): *An* encoding homomorphic wrt | and renaming transforms a symmetric solutions to the LEP in the source language into a symmetric solution to the LEP in the target language

The separation between π and π_I, ccs_{vp}

- **Theorem:** If a network with at least two nodes has a well-balanced automorphism $\sigma \neq id$ such that
  - $\forall i$ and $\forall$ node P, if $\sigma^i \neq id$ then there is no arc between P and $\sigma^i(P)$,

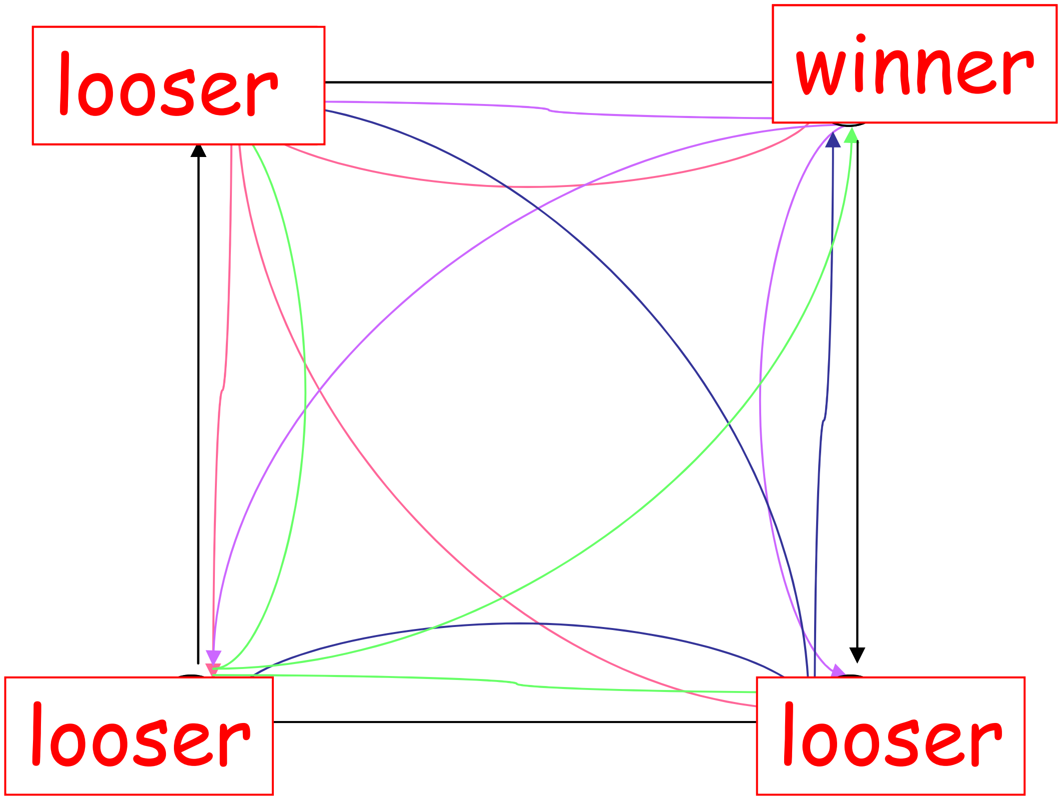
  then in π_I and ccs_{vp} there is no symmetric solution to the LEP.

- **Example**: a network which satisfies the above condition



The separation between π and π_I, ccs_{vp}

- A solution to the leader election problem for the same network in π



The separation between π and π_I, ccs_{vp}

- **Corollary:** there does not exists an encoding of π (π with mixed choice) in π_s which is homomorphic wrt | and renaming, does not increase the connectivity, and preserves the observables on every computation.