GETCO'2000

Occurrences Counting Analysis
for the π-calculus

Jérôme Feret
École normale supérieure
`http://www.di.ens.fr/~feret`

August 21, 2000

Overview

1. Mobile systems
   (a) What is it ?
   (b) π-calculus
   (c) Non-standard semantics

2. Approximate the collecting semantics
   (a) Abstract interpretation
   (b) Control flow analysis
   (c) Occurrences counting analysis
   (d) Examples

3. Trace-based analysis
   (a) Dynamic partitioning
   (b) Dead lock analysis
   (c) Example

Mobile systems

Mobile system

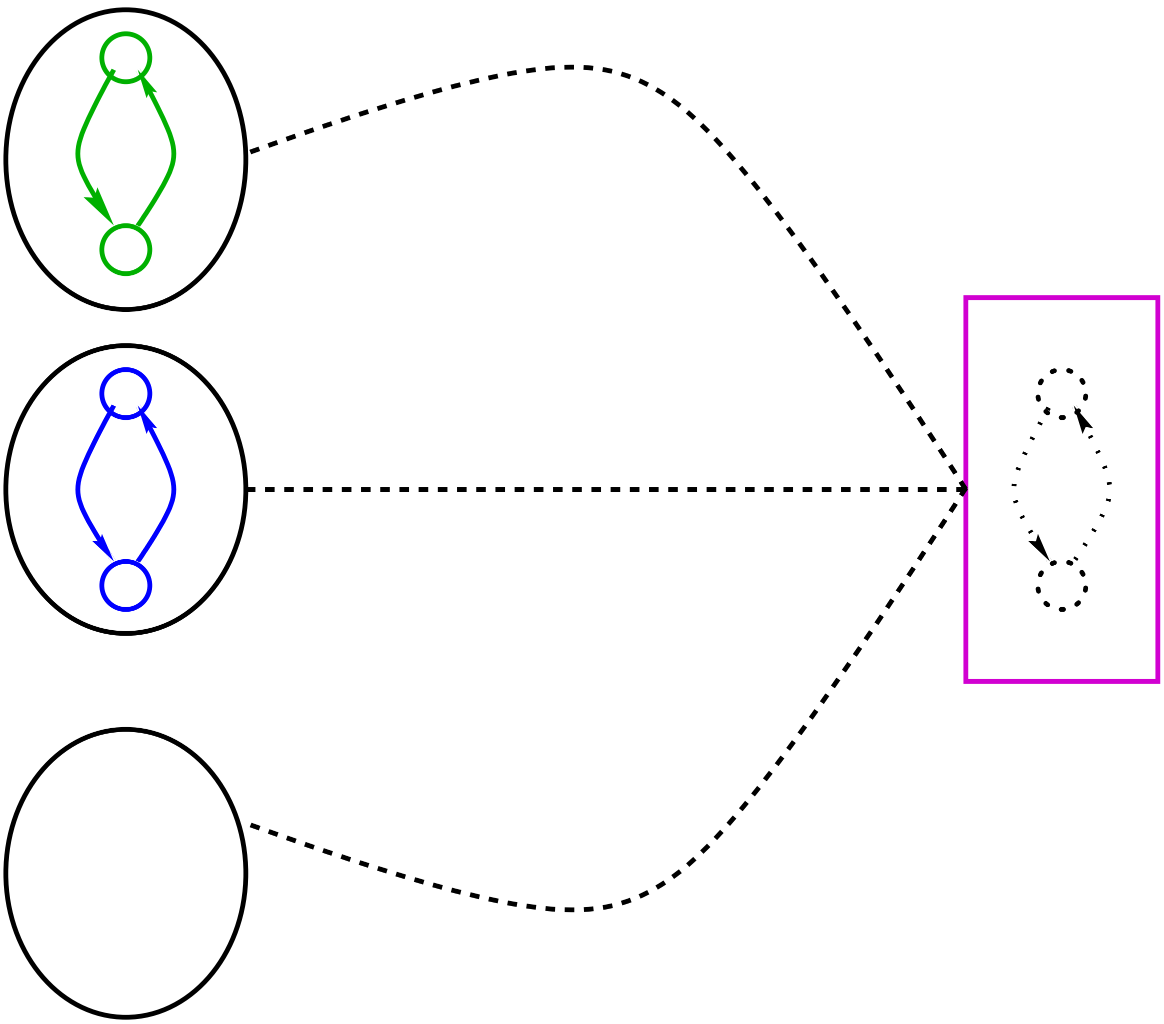
A pool of processes which interact via communications.

Communications allow to

- synchronize process computation;
- change structure of processes;
- create new communication links;
- create new processes.

Topology of interaction may be unbounded !

Example: a server



Objectives

We need a sound description of the multiset of the processes that occur inside computation sequences

- to prove that physical resources are not exhausted;

- to refine control flow analysis by detecting that some processes can never communicate,
  by detecting mutual exclusion;

- to provide a good criterion of partitioning,
  for dead lock analysis.

We propose a polynomial solution.

π-calculus : syntax

Let *Channel* be an infinite set of channel names, and *Label* an infinite set of labels,

$$\begin{array}{lcll} P & ::= & \text{action}.P & \text{(Action)} \\ & \mid & (P \mid P) & \text{(Parallel composition)} \\ & \mid & (P+P) & \text{(Non deterministic choice)} \\ & \mid & \emptyset & \text{(End of a process)} \end{array}$$

$$\begin{array}{lcll} \text{action} & ::= & c!^{i}[x_1, ..., x_n] & \text{(Message)} \\ & \mid & c?^{i}[x_1, ..., x_n] & \text{(Input guard)} \\ & \mid & *c?^{i}[x_1, ..., x_n] & \text{(Replication guard)} \\ & \mid & (\nu\ x) & \text{(Channel creation)} \end{array}$$

where $n \geqslant 0$,

$c, x_1, ..., x_n, x, \in$ *Channel* and $i \in$ *Label*.

ν and $?$ are the only name binders. We denote by $\mathcal{FN}(P)$ the set of free names in P, and by $\mathcal{BN}(P)$ the set of bound names in P.

Transition semantics

A reduction relation and a congruence relation give the semantics of the π-calculus:

- the reduction relation specifies the result of process computations:

$$
\begin{array}{rcl}
c?^{i}[\overline{y}]Q \mid c!^{j}[\overline{x}]P & \xrightarrow{i,j} & Q[\overline{y} \leftarrow \overline{x}] \mid P \\
*c?^{i}[\overline{y}]Q \mid c!^{j}[\overline{x}]P & \xrightarrow{i,j} & Q[\overline{y} \leftarrow \overline{x}] \mid *c?^{i}[\overline{y}]Q \mid P \\
P+Q & \xrightarrow{\varepsilon} & P \\
P+Q & \xrightarrow{\varepsilon} & Q
\end{array}
$$

- the congruence relation reveals redexs:
  - names renaming (α-conversion),
  - structural modifications
    (Commutativity, associativity, and so on).

Example: syntax

$$\mathcal{S} := (\nu\ \mathrm{port})$$
$$(\mathrm{Instance}\ \mid\ \mathrm{port}!^{5}[]\ \mid\ \mathrm{port}!^{6}[]\ \mid \mathrm{port}!^{7}[])$$

where

$$\mathrm{Instance} := *\mathrm{port}?^{0}[](\nu\ in)(\nu\ out)(\nu\ query)$$
$$(in!^{1}[query]$$
$$\mid\ in?^{2}[response].(out!^{3}\,[response]\ \mid\ \mathrm{port}!^{4}[]))$$

Example: computation

$$(\nu\ \mathrm{port})$$
$$(\mathrm{Instance} \mid \mathrm{port}!^{5}[] \mid \mathrm{port}!^{6}[] \mid \mathrm{port}!^{7}[])$$

$$\overset{(0,5)}{\longrightarrow}$$

$$(\nu\ \mathrm{port})(\nu\ in_1)(\nu\ out_1)(\nu\ query_1)$$
$$(\ \mathrm{Instance} \mid \mathrm{port}!^{6}[] \mid \mathrm{port}!^{7}[]$$
$$\mid in_1!^{1}[query_1]$$
$$\mid in_1?^{2}[response].(out_1!^{3}[response] \mid \mathrm{port}!^{4}[]))$$

$$\overset{(2,1)}{\longrightarrow}$$

$$(\nu\ \mathrm{port})(\nu\ in_1)(\nu\ out_1)(\nu\ query_1)$$
$$(\ \mathrm{Instance} \mid \mathrm{port}!^{4}[] \mid \mathrm{port}!^{6}[] \mid \mathrm{port}!^{7}[]$$
$$\mid out_1!^{3}[query_1])$$

Non-standard semantics

A refined semantics in where

- recursive instances of processes are identified with unambiguous markers;
- channel names are enriched with the marker of the process which has declared them.

Example: non-standard configuration

$$(\nu\ \mathrm{port})(\nu\ in_1)(\nu\ out_1)(\nu\ query_1)$$
$$(\mathrm{Instance}\ \mid\ \mathrm{port}!^{4}[]\ \mid\ \mathrm{port}!^{6}[]\ \mid\ \mathrm{port}!^{7}[]\ \mid\ out_1!^{3}[query_1])$$

$$\left\{\begin{array}{l}
\left(0, \varepsilon, \left\{\mathrm{port} \mapsto (\mathrm{port}, \varepsilon)\right.\right) \\
\left(3, id, \left\{\begin{array}{ll} out & \mapsto (out, id) \\ response & \mapsto (query, id) \end{array}\right.\right) \\
\left(4, id, \left\{\mathrm{port} \mapsto (\mathrm{port}, \varepsilon)\right.\right) \\
\left(6, \varepsilon, \left\{\mathrm{port} \mapsto (\mathrm{port}, \varepsilon)\right.\right) \\
\left(7, \varepsilon, \left\{\mathrm{port} \mapsto (\mathrm{port}, \varepsilon)\right.\right)
\end{array}\right\}$$

Marker allocation

Markers are binary trees:

- leaves are not labeled;
- nodes are labeled with a pair $(i, j) \in \mathit{Label}^2$.

They are recursively calculated when resources are fetched.

Coherence

Theorem: Standard semantics and non-standard semantics are bisimilar.

The proof mainly relies on the consistence of marker allocation.

Abstraction

Abstract interpretation

$(\mathcal{C}, C_0, \longrightarrow)$ is a transition system,

$$\mathcal{S} = \{C \mid \exists i \in C_0,\ i \longrightarrow^* C\} = lfp_{\emptyset}\mathbb{F}$$
$$\text{where } \mathbb{F}\ :\ X \mapsto C_0 \cup \{C' | \exists C \in X,\ C \rightarrow C'\}$$

- $(\wp(\mathcal{C}), \subseteq, \cup, \emptyset, \cap, \mathcal{C}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathcal{D}^\sharp, \sqsubseteq, \sqcup, \bot, \sqcap, \top)$
- an abstract transition relation $\rightsquigarrow$ on $\mathcal{D}^\sharp$

Coherence hypothesis:
If $C \in \gamma(C^\sharp)$ and $C \overset{\lambda}{\rightarrow} \overline{C}$, then there exists $\overline{C}^\sharp$ such that $C^\sharp \overset{\lambda}{\rightsquigarrow} \overline{C}^\sharp$ and $\overline{C} \in \gamma(\overline{C}^\sharp)$.

$$\begin{array}{ccc} C & \xrightarrow{\overset{\lambda}{\rightarrow}} & \overline{C} \\ \gamma\big\uparrow & & \gamma\big\uparrow \\ C^\sharp & \xrightarrow{\overset{\lambda}{\rightsquigarrow}} & \overline{C}^\sharp \end{array}$$

$$\mathcal{S} \subseteq \bigcup_{n \in \mathbb{N}} \gamma({\mathbb{F}^\sharp}^n(\bot))$$
$$\text{where } \mathbb{F}^\sharp(C^\sharp) = \alpha(C_0)) \sqcup C^\sharp \sqcup \left(\bigsqcup\{\overline{C^\sharp} \mid C^\sharp \rightsquigarrow \overline{C^\sharp}\}\right)$$

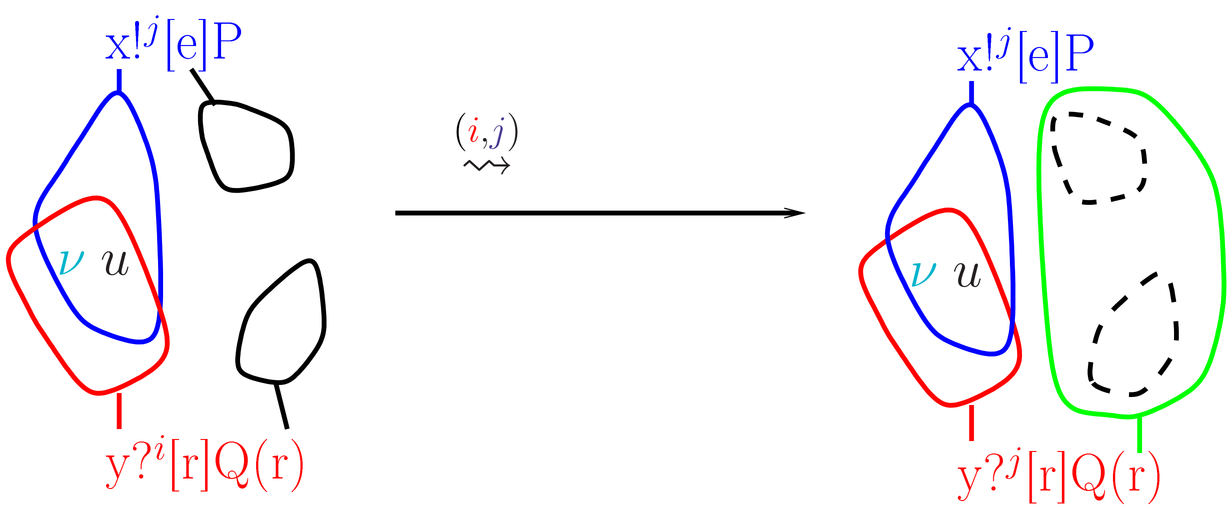
Control flow analysis

A description of the communication topology.

$$\left\{\begin{array}{l}
\left(0, \varepsilon, \left\{\text{port} \mapsto (\text{port}, \varepsilon)\right.\right) \\
\left(3, id, \left\{\begin{array}{ll} out & \mapsto (out, id) \\ response & \mapsto (query, id) \end{array}\right.\right) \\
\left(4, id, \left\{\text{port} \mapsto (\text{port}, \varepsilon)\right.\right) \\
\left(6, \varepsilon, \left\{\text{port} \mapsto (\text{port}, \varepsilon)\right.\right) \\
\left(7, \varepsilon, \left\{\text{port} \mapsto (\text{port}, \varepsilon)\right.\right)
\end{array}\right\}$$

$\Longrightarrow \{(\text{port},\text{port}),(out,out),(response,query)\}$

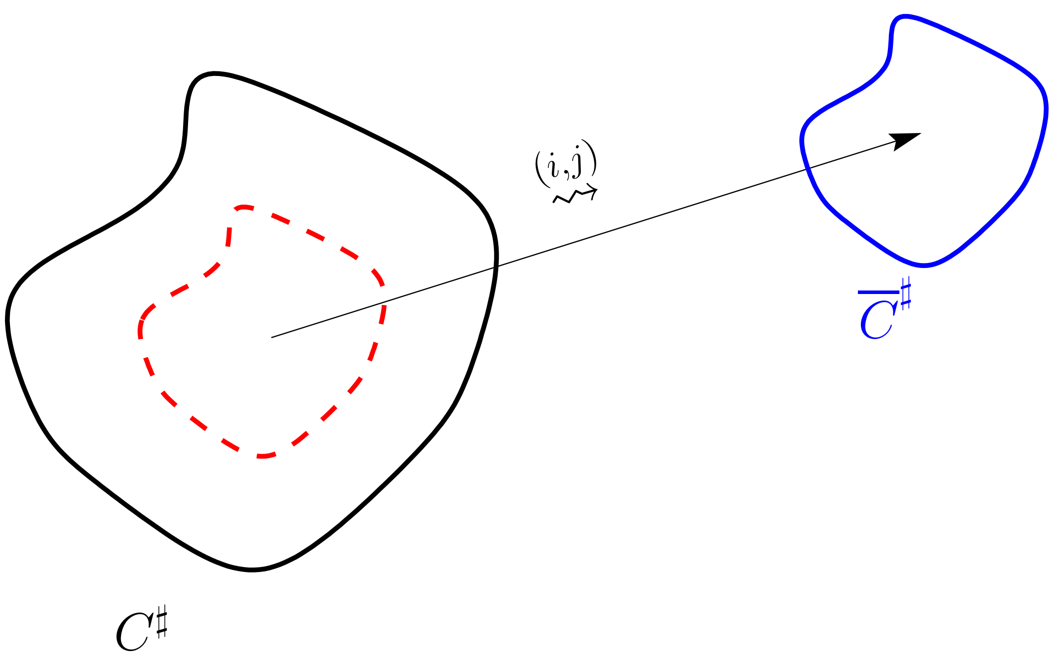
Abstract transition



Occurrences counting analysis

$$\left\{\begin{array}{l}\left(0, \varepsilon, \left\{\text{port} \quad \mapsto (\text{port}, \varepsilon)\right.\right)\\ \left(3, id, \left\{\begin{array}{ll} out & \mapsto (out, id)\\ response & \mapsto (query, id)\end{array}\right.\right)\\ \left(4, id, \left\{\text{port} \quad \mapsto (\text{port}, \varepsilon)\right.\right)\\ \left(6, \varepsilon, \left\{\text{port} \quad \mapsto (\text{port}, \varepsilon)\right.\right)\\ \left(7, \varepsilon, \left\{\text{port} \quad \mapsto (\text{port}, \varepsilon)\right.\right)\end{array}\right\}$$

Abstract transition



Abstract domains

We design a domain for representing numerical constrains between

- number of occurrences of processes $\sharp(i)$;
- number of performed transitions $\underline{\sharp}(i,j)$.

We use the product of

- a non-relational domain:
  $\Longrightarrow$ the interval lattice;
- a relational domain:
  $\Longrightarrow$ the lattice of affine relationships.

Interval narrowing

An exact reduction is exponential.

We use:

- Gaus reduction:

$$\begin{cases} x+y+z=1 \\ x+y+t=2 \end{cases} \Longrightarrow \begin{cases} x+y+z=1 \\ t-z=1 \end{cases}$$

- Interval propagation:

$$\begin{cases} x+y+z=3 \\ x \in [|0;\infty|[\\ y \in [|0;\infty|[\\ z \in [|0;\infty|[\end{cases} \Longrightarrow \begin{cases} x+y+z=3 \\ x \in [|0;3|] \\ y \in [|0;\infty|[\\ z \in [|0;\infty|[\end{cases}$$

- Redundancy introduction:

$$\begin{cases} x+y-z=3 \\ x \in [|1;2|[\end{cases} \Longrightarrow \begin{cases} x+y-z=3 \\ y-z \in [|1;2|] \\ x \in [|1;2|] \end{cases}$$

to get a polynomial approximated reduction.

Example: non-exhaustion of resources

$$
\begin{array}{l}
\mathcal{S} := (\nu\ \mathrm{port}) \\
\qquad (\mathrm{Instance}\ \mid\ \mathrm{port}!^{5}[]\ \mid\ \mathrm{port}!^{6}[]\ \mid \mathrm{port}!^{7}[])
\end{array}
$$

where

$$
\begin{array}{l}
\mathrm{Instance} := *\mathrm{port}?^{0}[](\nu\ in)(\nu\ out)(\nu\ query) \\
\qquad\qquad (in!^{1}[query] \\
\qquad\qquad \mid\ in?^{2}[response].(out!^{3}\ [response]\ \mid\ \mathrm{port}!^{4}[]))
\end{array}
$$

$$
\begin{cases}
\sharp(0) = 1 \\
\sharp(3) \in [|0; \infty|[\\
\sharp(i) \in [|0; 3|],\ \ \forall i \in \{1; 2; 4\} \\
\sharp(i) \in [|0; 1|],\ \ \forall i \in [|5; 7|] \\
\sharp(1) + \sharp(4) + \sharp(5) + \sharp(6) + \sharp(7) = 3 \\
\sharp(1) = \sharp(2)
\end{cases}
$$

Example: exhaustion of resources

$$\mathcal{S} := (\nu\ \mathrm{port})$$
$$(\mathrm{Instance}\ \mid\ \mathrm{port}!^{5}[]\ \mid\ \mathrm{port}!^{6}[]\ \mid \mathrm{port}!^{7}[])$$

where

$$\mathrm{Instance} := *\mathrm{port}?^{0}[](\nu\ in)(\nu\ out)(\nu\ query)$$
$$(in!^{1}[query]$$
$$\mid\ in?^{2}[response].out!^{3}\ [response]$$
$$\mid\ \mathrm{port}!^{4}[])$$

$$\begin{cases} \sharp(0) = 1 \\ \sharp(i) \in [|0;\infty|[,\ \forall i \in \{1;2;3;4\} \\ \sharp i \in [|0;1|],\ \forall i \in \{5;6;7\} \\ \sharp(1) + \sharp(3) = \sum_{i \in \{4,5,6,7\}} \underline{\sharp}(0,i) \end{cases}$$

Example: mutual exclusion

$$\begin{aligned}
A &:= *a?^1[x](x!^2[a] + c?^3[u]d!^4[u]) \\
B &:= *b?^5[x](x!^6[b] + c!^7[e]\ ) \\
C &:= a!^8[b] \\
P &:= A \mid B \mid C
\end{aligned}$$

$\Longrightarrow$ We detect that processes 3 and 7 never communicate.

since the following system:

$$\begin{cases}
\sharp(2) + \sharp(3) + \sharp(6) + \sharp(7) + \sharp(8) = 1 \\
\sharp(3) \in [|1;\infty|[\\
\sharp(7) \in [|1;\infty|[
\end{cases}$$

has no solution in $\mathbb{N}^+$.

Trace-based analysis

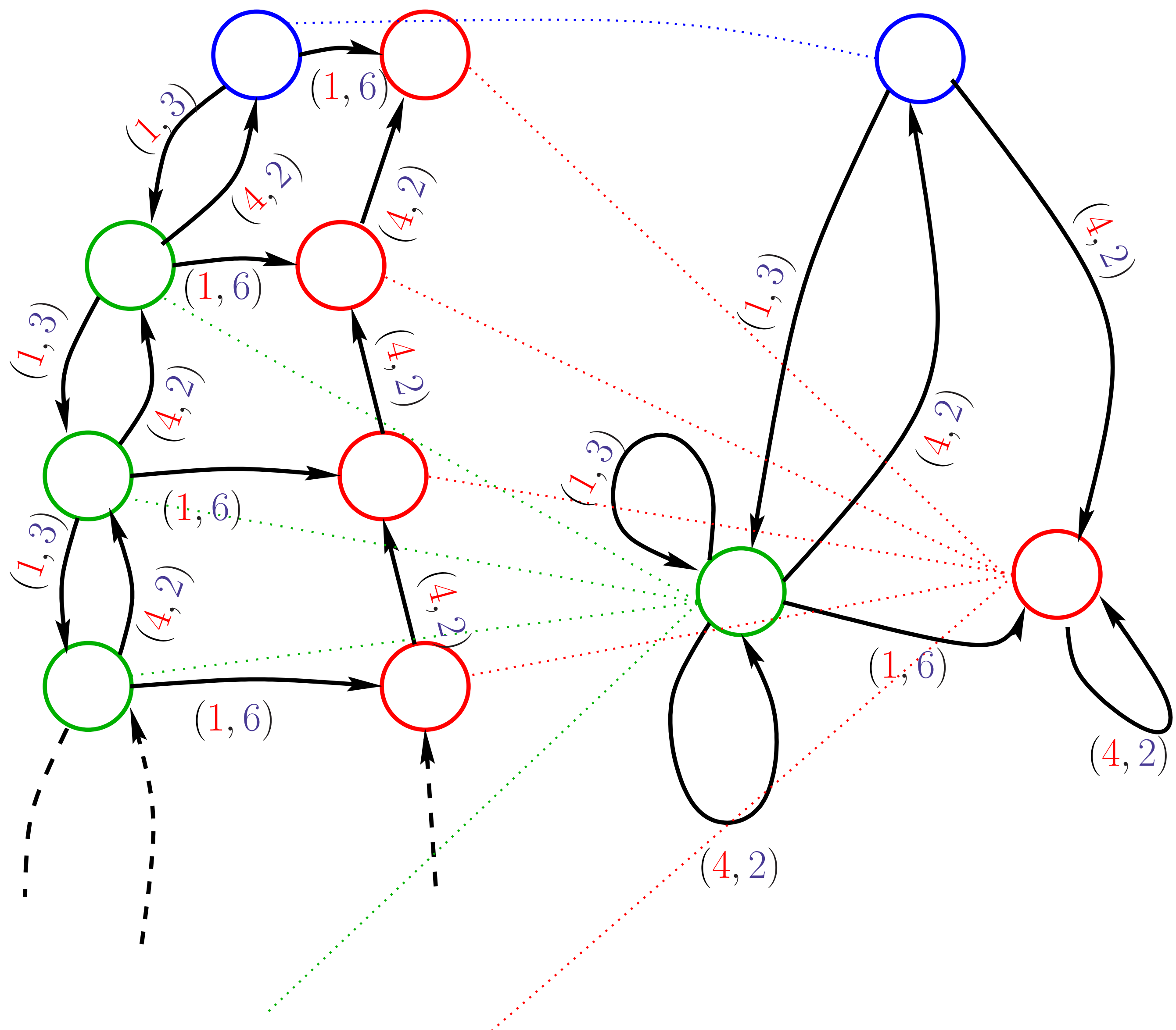
Main idea

We want to approximate the set of the configurations by which no infinite computation sequence can pass.

We propose to

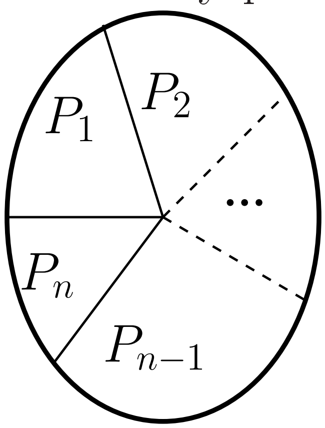
1. abstract the trace semantics of a mobile system;
2. for each configuration,
   - approximate the set of the transitions that may occur inside a computation sequence which stems from this configuration;
   - detect and prove whether this set defines a well-founded relation.

Quotient



Partitioning

We finitely partition $\mathcal{D}^\sharp$



$$\begin{cases} \forall k \in [|1;n|],\ P_k \neq \bot \\ \forall k,l \in [|1;n|],\ k \neq l \implies P_k \sqcap P_l = \bot \\ \bigsqcup (P_k)_{k \in [|1;n|]} = \top \end{cases}$$

$\mathcal{D}^\sharp$

by using our occurrences counting analysis.

We iteratively construct both

- a transition system over (P_i),
- a representation function $f : [|1;n|] \rightarrow \mathcal{D}^\sharp$:

If $P_k \sqcap f(P_k) \overset{(i,j)}{\rightsquigarrow} \overline{C}^\sharp$ with $\overline{C}^\sharp \sqcap P_l \neq \bot$

$$\text{then } \begin{cases} f(P_l) \leftarrow f(P_l) \sqcup (\overline{C}^\sharp \sqcap P_l) \\ \text{the transition } P_k \overset{(i,j)}{\rightarrow} P_l \text{ is added} \end{cases}$$

Proof of termination

How to check that transition systems are well-founded?

Abstracting environments away,
transition rules look like chemical reactions.

$$A|B \rightarrow A_1|A_2|\ldots|B_1|B_2|\ldots \qquad \text{(communication)}$$
$$C|D \rightarrow C|C_1|C_2|\ldots|D_1|D_2|\ldots \qquad \text{(resource fetching)}$$

We decompose each transition in two half-transitions:

communication	resource fetching
$A \rightarrow A_1\|A_2\|\ldots$	$C \rightarrow C$
$B \rightarrow B_1\|B_2\|\ldots$	$D \rightarrow C_1\|C_2\|\ldots\|D_1\|D_2\|\ldots$

Then we check if the following relation is well-founded:

communication	resource fetching
$A > A_1,\ A > A_2,\ \ldots$	$D > C_1,\ D > C_2,\ \ldots$
$B > B_1,\ B > B_2,\ \ldots$	$D > D_1,\ D > D_2,\ \ldots$

Example: a stack

$$
\begin{aligned}
\mathcal{S} \;:=\; & (\nu\ \text{push})(\nu\ \text{pop}) \\
& ((*\text{push?}^1[](\text{pop!}^2[] \mid \text{push!}^3[])) \\
& \mid *\text{pop?}^4[] \\
& \mid *\text{push?}^5[] \\
& \mid \text{push!}^6[])
\end{aligned}
$$

$$
\begin{cases}
\pi(1) = 1,\ \pi(2) \in [|0;+\infty|[,\ \pi(3) \in [|0;1|], \\
\pi(4) = 1,\ \pi(5) = 1,\ \pi(6) \in [|0;1|], \\
\underline{\pi}(1,6) \in [|0;1|],\ \underline{\pi}(5,3) \in [|0;1|],\ \underline{\pi}(5,6) \in [|0;1|], \\
\underline{\pi}(1,3) \in [|0;\infty|[,\ \underline{\pi}(4,2) \in [|0;\infty|[.
\end{cases}
$$

The analysis has proved that the computations of our system are bound to terminate as soon as a communication $(5,3)$ or a communication $(5,6)$ is performed.

Conclusion

- Our framework allows to infer a sound uniform description of mobile systems in the π-calculus.
- It has succeeded in proving:
  - non-exhaustion, in a polynomial time;
  - mutual exclusion, in a polynomial time;
  - some dead locks, in an exponential time.

Future Works

- Refining our initial partitioning;
- investigating other approximated algorithms;
- designing a modular analysis;
- including administrative sites (*mobile ambients*).

$\Longrightarrow$ To analyze big programs.