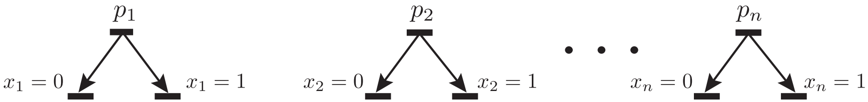
Soundness in negotiations

Igor Walukiewicz

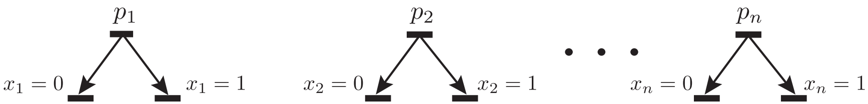
CNRS, Bordeaux University

Joint work with
Javier Esparza, Denis Kuperberg, and Anca Muscholl

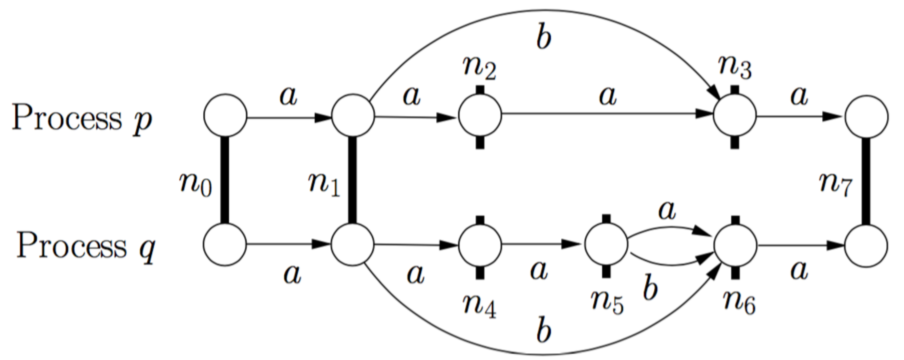
Verification of concurrent systems suffers from the state explosion problem.



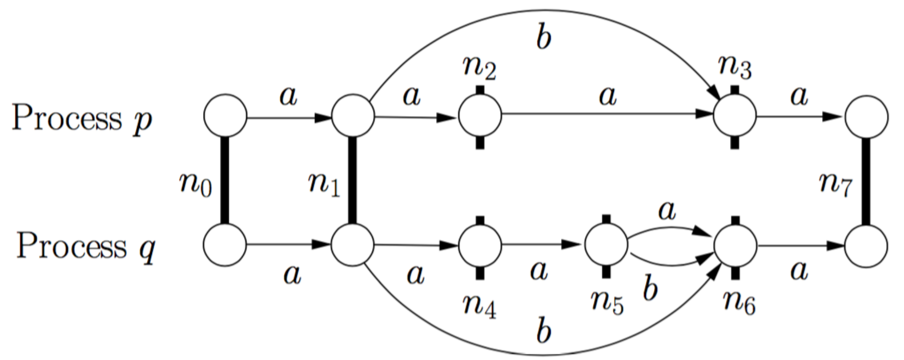
Verification of concurrent systems suffers from the state explosion problem.



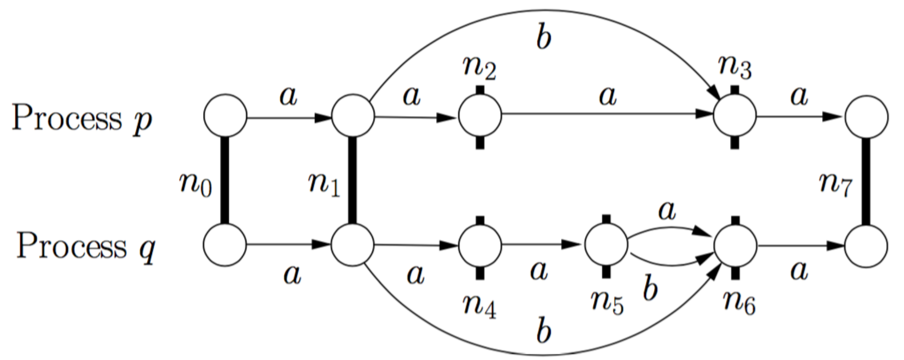
Negotiations is a restricted model for which some verification problems are much easier than usually.



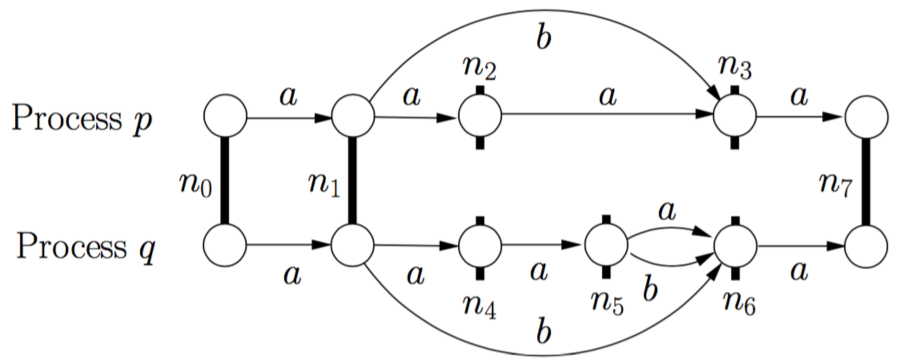
- $Proc$: processes.
- N : atomic negotiations (nodes); $dom : N \to \mathcal{P}(Proc)$.
- R : outcomes.
- $\delta : N \times R \times P \stackrel{\cdot}{\longrightarrow} \mathcal{P}(N)$: partial transition function
  $\delta(n, a, p)$ is a set of next atomic negotiations for process p;
  for every n, $a \in out(n)$, $p \in dom(n)$,



- A configuration $C : Proc \to \mathcal{P}(N)$
- n is enabled in C if $n \in C(p)$ for all $p \in dom(n)$.
- A run $C_1 \stackrel{(n_1,a_1)}{\longrightarrow} C_2 \stackrel{(n_2,a_2)}{\longrightarrow} C_3 \ldots$
- A successful run $C_{init} \stackrel{w}{\longrightarrow} C_{fin}$



- A negotiation is **sound** if every run $C_{init} \xrightarrow{w} C$ can be completed to a successful run.



- **Deterministic negotiation:** $\delta(n, a, p)$ is at most singleton.
- **Graph of a negotiation** (see above).
- **Local path** a path in the graph of a negotiation.
- **Acyclic negotiation** when its graph is acyclic.

Rem: For acyclic negotiations: sound $\equiv$ no-deadlock.

Soundness: every run can be completed to a successful run

Deterministic acyclic negotiations

- Soundness in NLOGSPACE
- $L(N) \cap L(A) \neq \emptyset$ NP-complete

A local path $n_0 \xrightarrow{p_0,a_0} n_1 \xrightarrow{p_1,a_1} \dots \xrightarrow{p_{k-1},a_{k-1}} n_k$ is **realizable** if it is a part of a run.

Lemma

Every local path is realizable.

Proof

Atomic negotiation n_0 is enabled in C_{init}.

Suppose n_i is enabled in C_i.
Let C'_i be the result of executing a_i. We have $C'_i(p) = n_{i+1}$
By soundness from C'_i we can reach C_{fin}.
So on the way we reach C_{i+1} where n_{i+1} is enabled.

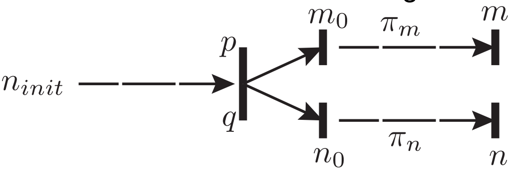
A local path $n_0 \xrightarrow{p_0,a_0} n_1 \xrightarrow{p_1,a_1} \dots \xrightarrow{p_{k-1},a_{k-1}} n_k$ is **realizable** if it is a part of a run.

Lemma

Every local path is realizable.

Lemma

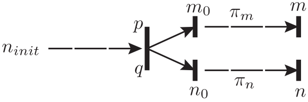
There is an execution containing m and n iff there is a pattern:



$\pi_m \cap \pi_n = \emptyset$

Lemma

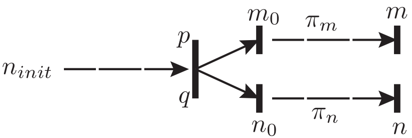
There is an execution containing m and n iff there is a pattern:



$\pi_m \cap \pi_n = \emptyset$

Lemma

Acyclic $\mathcal{N}$ is not sound iff its graph has a pattern:



$q \in dom(m)$

$m \preccurlyeq n$

$\pi_m \cap \pi_n = \emptyset$

Theorem

Soundness of acyclic deterministic negotiations is NLOGSPACE-complete.

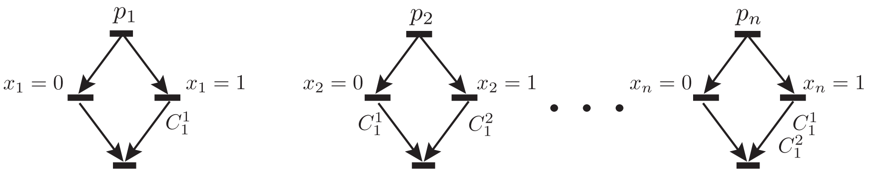
Not everything is easy to check for deterministic acyclic negotiations

Thm

$L(\mathcal{N}) \cap L(\mathcal{A}) \neq \emptyset$ is NP-complete, for $\mathcal{N}$ an acyclic deterministic negotiation and $\mathcal{A}$ a deterministic finite automaton.

1 in 3 SAT

$$(x_1 \vee \overline{x_2} \vee x_n) \wedge (x_2 \vee \overline{x_4} \vee x_n) \wedge \ldots$$



$$L(\mathcal{A}) = \{C_1^{i_1} C_2^{i_2} \ldots C_k^{i_k} : i_1, \ldots, i_k \in [n]\}$$

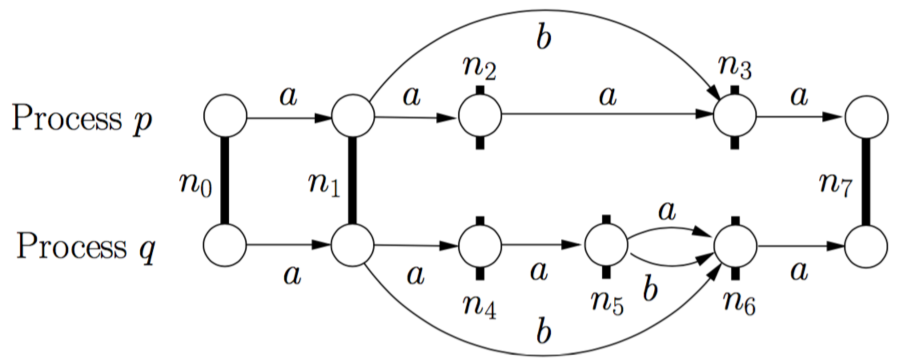
Soundness: every run can be completed to a successful run

Deterministic acyclic negotiations

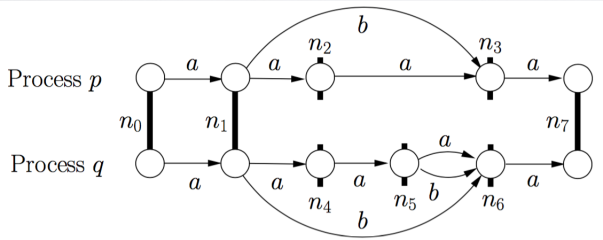
- Soundness in NLOGSPACE
- $L(N) \cap L(A) \neq \emptyset$ NP-complete

Verifying properties of sound acyclic deterministic negotiations

- some properties can be decided in PTIME
- races can be decided in PTIME



Atomic negotiations may have outcomes:
$alloc(x)$, $read(x)$, $write(x)$, and $dealloc(x)$.



(1) *Inconsistent data*: an atomic negotiation reads or writes a variable x while another atomic negotiation is writing, allocating, or deallocating it in parallel.

(2) *Never destroyed*: there is an execution in which a variable is allocated and then never deallocated before the execution ends.

(3) *Weakly redundant data*: there is an execution in which a variable is written and never read before it is deallocated or the execution ends.

(1) *Inconsistent data*: an atomic negotiation reads or writes a variable x while another atomic negotiation is writing, allocating, or deallocating it in parallel.
(2) *Never destroyed*: there is an execution in which a variable is allocated and then never deallocated before the execution ends.
(3) *Weakly redundant data*: there is an execution in which a variable is written and never read before it is deallocated or the execution ends.

Thm

These properties can be checked in PTIME for acyclic, deterministic, sound negotiations.

Concurrency of two actions

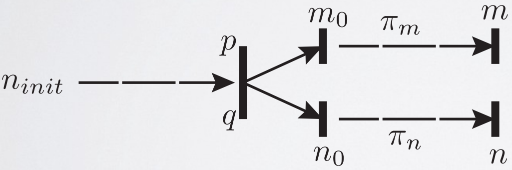
We write $m \parallel n$ if $\mathcal{N}$ has a reachable configuration C where both m and n are enabled.

Thm

We can decide in a linear time if in a given acyclic, deterministic, sound negotiation the two given atomic negotiations m, n satisfy $m \parallel n$.

Proposition

$m \parallel n$ iff there is a run containing m, n, and there is no local path from m to n or vice versa.



$$\pi_m \cap \pi_n = \emptyset$$

Thm [Kovalyov, Esparza]

For all deterministic negotiations there is a cubic algorithm for this problem.

Soundness: every run can be completed to a successful run

Deterministic acyclic negotiations

- Soundness in NLOGSPACE
- $L(N) \cap L(A) \neq \emptyset$ NP-complete

Verifying properties of sound acyclic deterministic negotiations

- some properties can be decided in PTIME
- races can be decided in PTIME

Soundness for bigger classes

- for weakly deterministic acyclic in PTIME
- without acyclicity coNP-hard

Thm [Espaza, Desel]

Soundness is PSPACE-complete for non-deterministic negotiations.
It is coNP-complete when they are acyclic.

Thm [Esparza, Desel]

Soundness is in PTIME for deterministic negotiations.

Thm

Soundness is in PTIME for acyclic weakly non-deterministic negotiations.

Thm

Soundness is coNP-complete for very weakly non-deterministic negotiations.

A process p is deterministic if $\delta(n, a, p)$ is at most a singleton, for all n, a.

A negotiation is weakly non-deterministic if for every $n \in N$ at least one of the processes in $dom(n)$ is deterministic.

Thm

Soundness can be decided in PTIME for acyclic, weakly non-deterministic negotiations.

A negotiation is weakly non-deterministic if for every $n \in N$ at least one of the processes in $dom(n)$ is deterministic.

Lemma

An acyclic weakly non-deterministic negotiation $\mathcal{N}$ is not sound if and only if:

- either its restriction $\mathcal{N}_D$ to deterministic processes is not sound,
- or, for some non-deterministic process p, its restriction to p and the deterministic processes is not sound.

Thm (Omitting)

It can be decided in PTIME if for a given deterministic, acyclic, and sound negotiation $\mathcal{N}$ and a set $B \subseteq N$ there is a successful run of $\mathcal{N}$ omitting B.

A negotiation is weakly non-deterministic if for every $n \in N$ at least one of the processes in $dom(n)$ is deterministic.

A negotiation is very weakly non-deterministic if for every $n \in N$ $a \in R$ and $p \in Proc$ there is a deterministic process q such that $q \in dom(n')$ for all $n' \in \delta(n, a, p)$. (q decides about the next negotiation)

det-acyclic: restriction to deterministic processes is acyclic.

Thm

Soundness of det-acyclic, very weakly non-deterministic negotiations is coNP-complete.

Soundness: every run can be completed to a successful run

Deterministic acyclic negotiations

- Soundness in NLOGSPACE
- $L(N) \cap L(A) \neq \emptyset$ NP-complete

Verifying properties of sound acyclic deterministic negotiations

- some properties can be decided in PTIME
- races can be decided in PTIME

Soundness for bigger classes

- for weakly deterministic acyclic in PTIME
- without acyclicity coNP-hard