Checking Partial-Order Properties of VASS

Florent Avellaneda
joint work with Rémi Morin (PhD advisor)

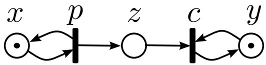
Laboratoire d'Informatique Fondamentale de Marseille, AMU & CNRS, UMR 7279

9 juillet 2013

Petri Net

A Petri net is a quadruple $\mathcal{N} = (P, T, W, \mu_{in})$ where :

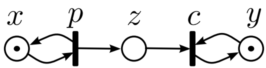
- P is a finite set places, and T is a finite set of transitions such that $P \cap T = \emptyset$.
- W is a map from $(P \times T) \cup (T \times P)$ to $\mathbb{N}$.
- μ_{in} is a map from P to $\mathbb{N}$, called the initial marking.



Petri Net

A Petri net is a quadruple $\mathcal{N} = (P, T, W, \mu_{in})$ where :

- P is a finite set places, and T is a finite set of transitions such that $P \cap T = \emptyset$.
- W is a map from $(P \times T) \cup (T \times P)$ to $\mathbb{N}$.
- μ_{in} is a map from P to $\mathbb{N}$, called the initial marking.



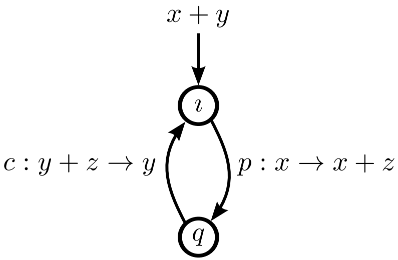
Two rules :

- $p : x \to x + z$
- $c : y + z \to y$

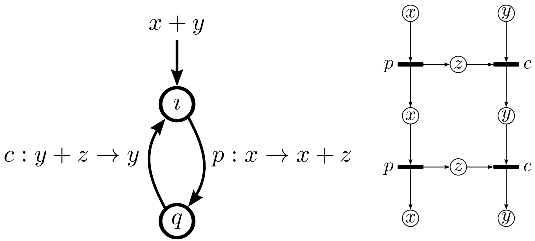
Petri Nets with States

A Petri Net with States (PNS) is an automaton $\mathcal{S} = (Q, \imath, \rightarrow, \mu_{in})$ where :

- Q is a finite set of states.
- $\imath \in Q$ is an initial state.
- $\rightarrow \subseteq Q \times R \times Q$ is a finite set of arcs labeled by rules.
- $\mu_{in} \in \mathbb{N}^P$ is a initial marking.



Partial-Order semantics (Process semantics)



A process of the computation sequence *pcpc*.

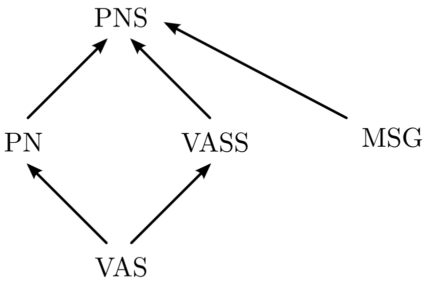
- A sequence can not be firable.
- A sequence may correspond to several processes.
- A process may correspond to several sequences.

References

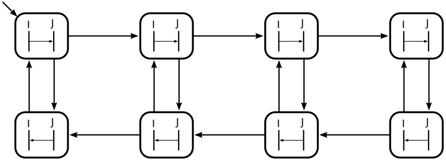
- **J. Engelfriet**. Branching processes of Petri nets. Acta Informatica 1991.
- **U. Goltz and W. Reisig**. The non-sequential behavior of Petri nets. Information and Control 1983.
- **W. Vogler**. Modular Construction and Partial Order Semantics of Petri Nets. Lecture Notes in Computer Science 1992.

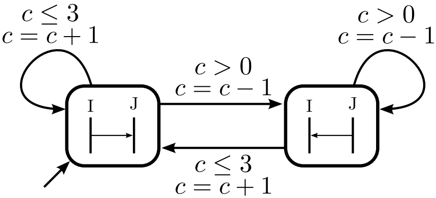
Some equivalences :

- Petri Nets with pure rules = VAS
- PNS with one state = Petri Nets
- PNS with pure rules = VASS
- MSG are PNS with special rules and FIFO semantics.



A Setting for MSG



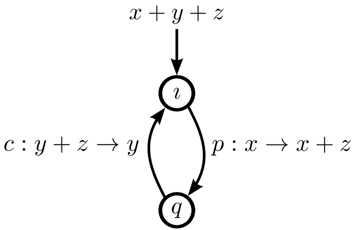


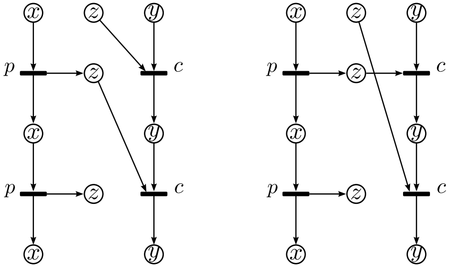
New features :

- Counters.
- Timers.
- Clocks.
- Dynamic creation of processes.

Notations

- $CS(\mathcal{S})$: all computation sequences of $\mathcal{S}$.
- $[\![u]\!]$: all processes of a sequence u.
- $[\![\mathcal{S}]\!]_\mu$: all processes of $\mathcal{S}$ from μ.





Two processes of the computation sequence *pcpc* from $x + y + z$

Inclusion Problem

Input $\mathcal{S}_1, \mathcal{S}_2$ two PNS with initial marking μ_1 and μ_2.

Question $[\![\mathcal{S}_1]\!]_{\mu_1} \subseteq [\![\mathcal{S}_2]\!]_{\mu_2}$?

This question is :

- decidable if $\mathcal{S}_1$ and $\mathcal{S}_2$ are Petri Nets,
- undecidable in general,
- undecidable even for bounded VASS.

Petri Nets are not equivalent to VASS !

1. Checking reachability properties
2. Checking MSO Properties
3. Conclusion

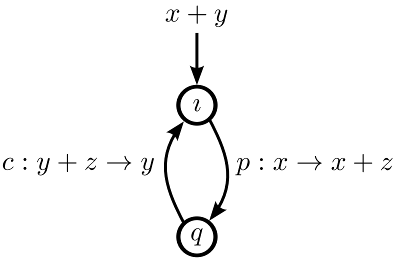
Plan

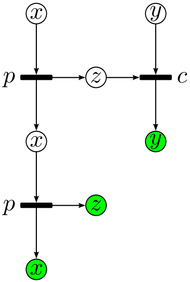
1. Checking reachability properties
2. Checking MSO Properties
3. Conclusion

Definition

Reachable

A marking μ is reachable in a PNS $\mathcal{S}$ if there exists a process of $\mathcal{S}$ which leads to the marking μ.



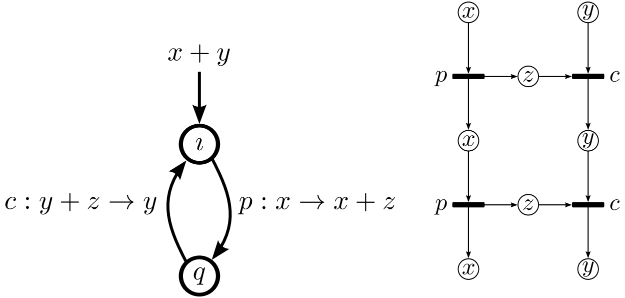


$x + z + y$ is reachable.

Definition

Prefix-reachable

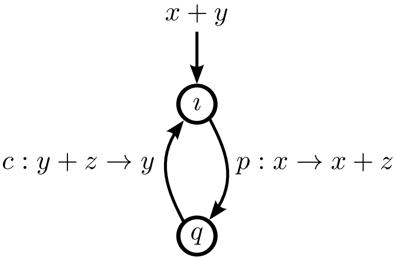
A marking μ is prefix-reachable in a PNS $\mathcal{S}$ if there exists a prefix of a process of $\mathcal{S}$ which leads to the marking μ.

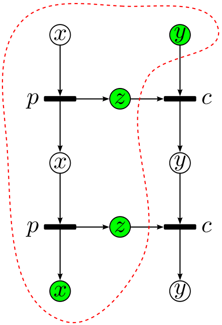


Definition

Prefix-reachable

A marking μ is prefix-reachable in a PNS $\mathcal{S}$ if there exists a prefix of a process of $\mathcal{S}$ which leads to the marking μ.



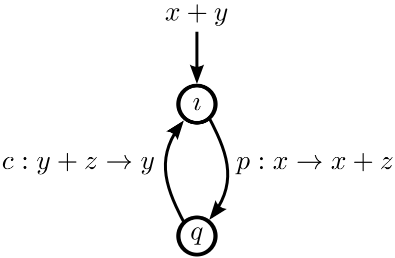


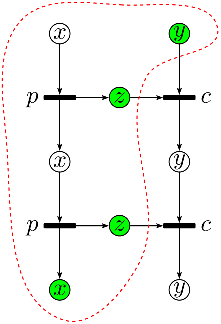
$x + 2 \cdot z + y$ is prefix-reachable.

Definition

Prefix-reachable

A marking μ is prefix-reachable in a PNS $\mathcal{S}$ if there exists a prefix of a process of $\mathcal{S}$ which leads to the marking μ.





$x + 2 \cdot z + y$ is prefix-reachable.
$x + n \cdot z + y$ is prefix-reachable.

Problem

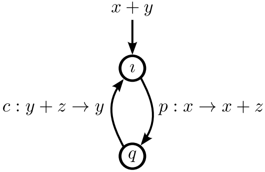
Prefix-Reachability Problem

Input: PNS $\mathcal{S}$, marking μ

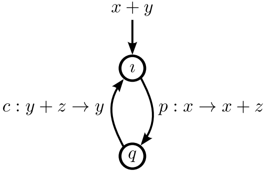
Question: μ is prefix-reachable ?

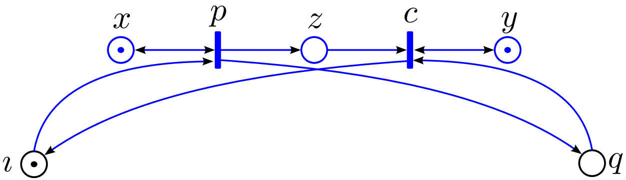
We prove that the prefix-reachability problem is decidable by reduction to the reachability problem of Petri nets.

From a PNS $\mathcal{S}$ to a Petri net $\mathcal{N}$

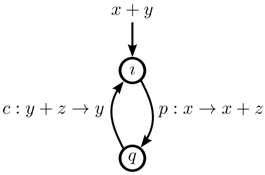


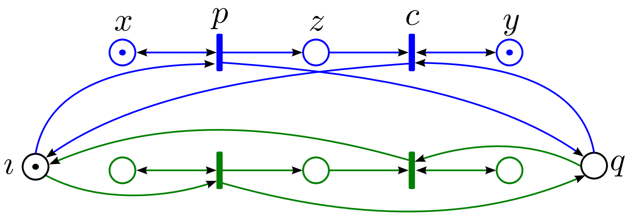
From a PNS $\mathcal{S}$ to a Petri net $\mathcal{N}$



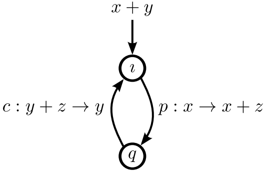


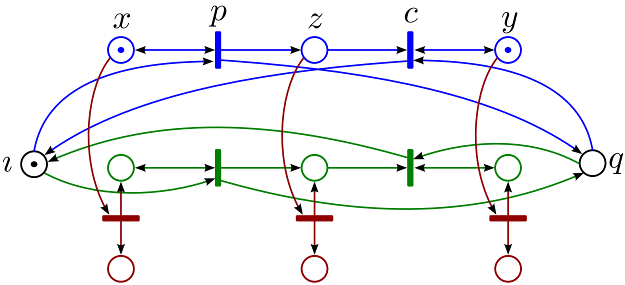
From a PNS $\mathcal{S}$ to a Petri net $\mathcal{N}$



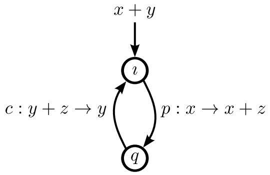


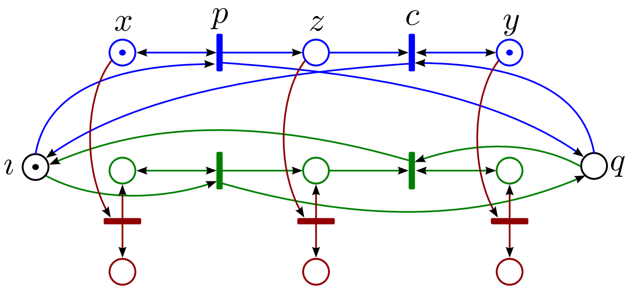
From a PNS $\mathcal{S}$ to a Petri net $\mathcal{N}$





From a PNS $\mathcal{S}$ to a Petri net $\mathcal{N}$

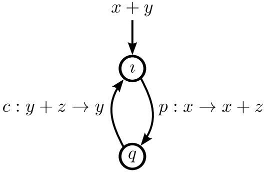


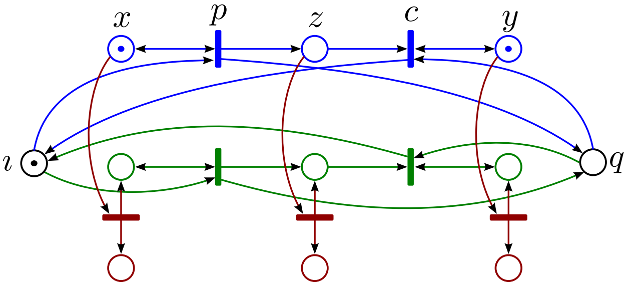


Theorem

A multiset of places $\mu \in \mathbb{N}^P$ is prefix-reachable in $\mathcal{S}$ if and only if there exists some reachable marking μ' in $\mathcal{N}$ such that $\mu = \mu'_{pre} + \mu'_{cut}$.

From a PNS $\mathcal{S}$ to a Petri net $\mathcal{N}$



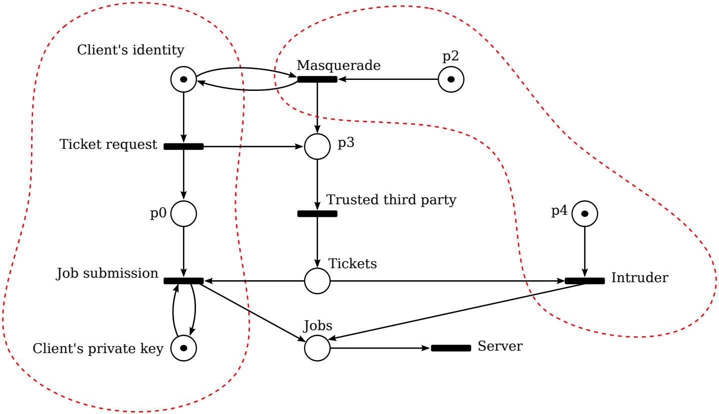


Corollary

Prefix-reachability, prefix-boundedness and prefix-covering can be solved by the same reduction.

Plan

1. Checking reachability properties
2. Checking MSO Properties
3. Conclusion



Three basic properties :

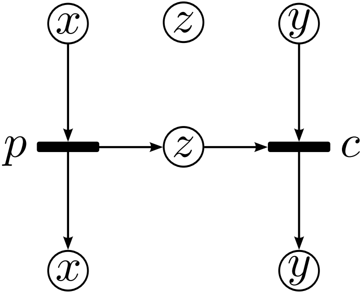
P1. A ticket cannot be consumed without the client's private key.

P2. The server does not consume jobs submitted by the intruder.

P3. The client consumes only tickets that it has requested.

Ideas

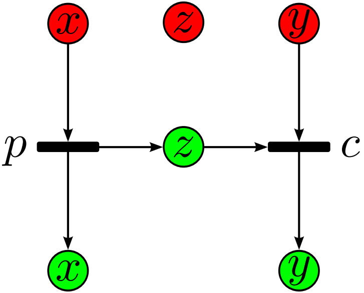
- Use words rather than partial orders.



Representative word :
$xyz\bar{x}pzx\bar{y}\bar{z}cy$

Ideas

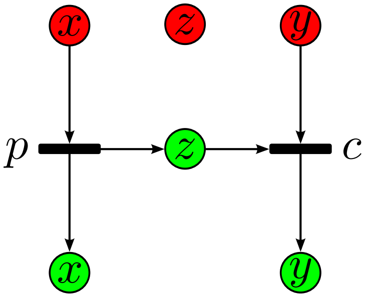
- Use words rather than partial orders.
- Color to encode the order.



Representative word :
$xyz\bar{x}pzx\bar{y}\bar{z}cy$

Ideas

- Use words rather than partial orders.
- Color to encode the order.



Representative word :
$xyz\bar{x}pzx\bar{y}\bar{z}cy$

There is exactly one process for each colored word.

Ideas

If $\mathcal{S}$ is bounded, we can unfold $\mathcal{S}$ to an automaton $\mathcal{S}'$ generating exactly the words representing the processes of $\mathcal{S}$.

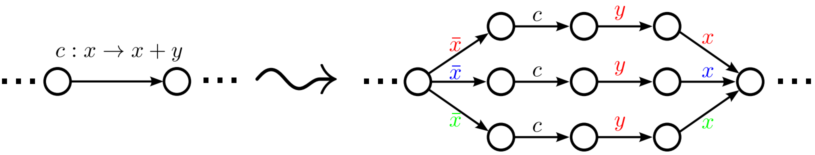
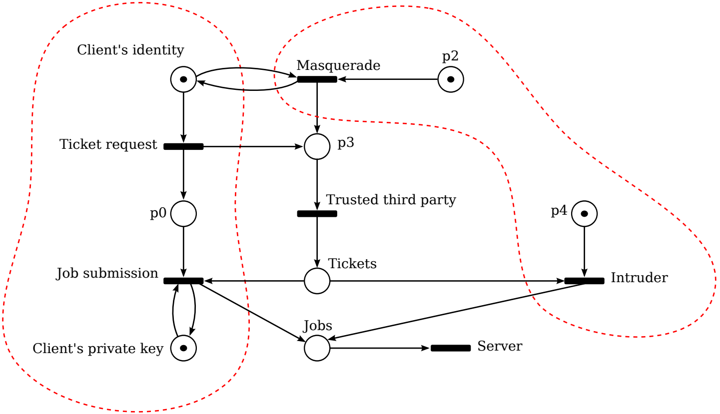


Figure: Unfolding of $\mathcal{S}$ to $\mathcal{S}'$.

Theorem

Let $\mathcal{S}$ be a bounded PNS and ψ be an MSO sentence over causal nets. Then $\mathcal{S} \models \psi$ is decidable.



Three basic properties :

P1. A ticket cannot be consumed without the client's private key.

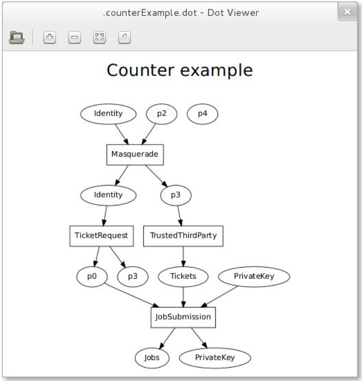
P2. The server does not consume jobs submitted by the intruder.

P3. The client consumes only tickets that it has requested.

We have implemented our technique on top of the tool MONA.

Results for this example :

- $P1 \Rightarrow P2$
- $P1 \not\Rightarrow P3$



Plan

1. Checking reachability properties
2. Checking MSO Properties
3. Conclusion

Summary

1. We introduce a natural partial order semantics for vector addition systems with states that extend the classical processes semantics of Petri Nets.

2. Basic problems about the set of markings reached along the processes can be reduced to the analogous problems for (unbounded) Petri nets.

3. We present a method to check any MSO property of processes for bounded PNS (not necesserily prefix-bounded).

Perspectives

1. Adapt the tool for MSG :
   - Consider the FIFO representation.
   - Rules in the states.
   - Merge with previous MSG analysis tools (ACSD'11).
2. Define a class of PNS "Globally Cooperative" :
   - Processes are MSO-definable.
   - Inclusion becomes decidable.
   - Reduce the complexity for basic problems.

Thank you for your attention