Reconfiguration in Stochastic Petri Nets

PhD Student: Samir Tigane
Supervisors: Laïd Kahloul and Souheib Baarir

LINFI Laboratory Biskra University, Algeria
Invited member at LIP6.

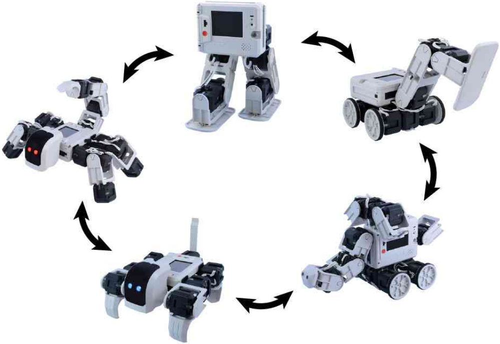
MoVe
October 11 2019

Plan

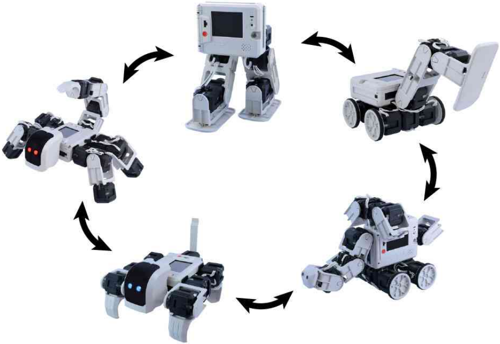
1. Motivation Behind Reconfiguration in PNs
2. Contribution: Reconfigurable Generalized Stochastic PNs
3. Comparative Evaluation
4. Conclusion and Perspectives

Why we consider reconfigurability in PNs?

- Modern DESs are more likely to be structurally dynamic and variably interconnected at run-time.



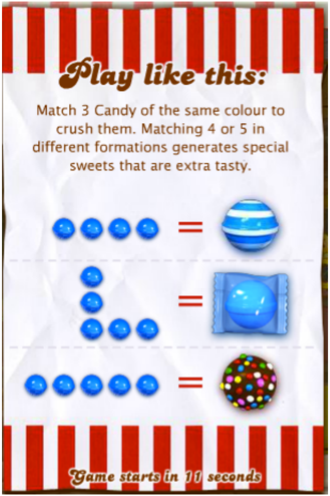
- Modern DESs are more likely to be structurally dynamic and variably interconnected at run-time.



- PNs are characterized by their rigid structures.

- Reconfiguration is expressed via transformation rules

- Reconfiguration is expressed via transformation rules



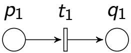
Play like this:

Match 3 Candy of the same colour to crush them. Matching 4 or 5 in different formations generates special sweets that are extra tasty.

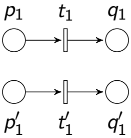
Game starts in 11 seconds

Transformation rule

Left-hand Side

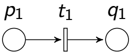


Right-hand Side

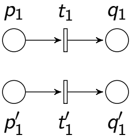


Transformation rule

Left-hand Side

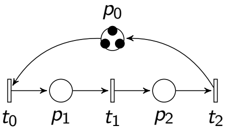


Right-hand Side



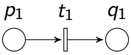
Rule application

Initial Configuration

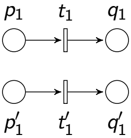


Transformation rule

Left-hand Side

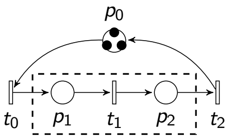


Right-hand Side



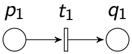
Rule application

Initial Configuration

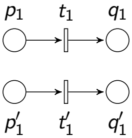


Transformation rule

Left-hand Side

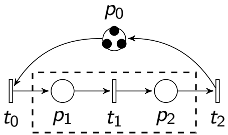


Right-hand Side

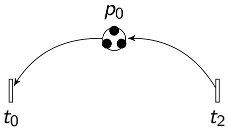


Rule application

Initial Configuration

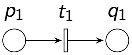


Context Graph

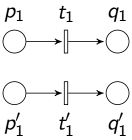


Transformation rule

Left-hand Side

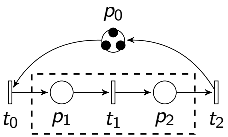


Right-hand Side

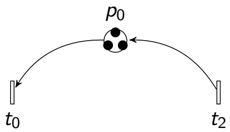


Rule application

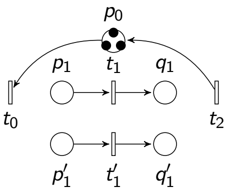
Initial Configuration



Context Graph

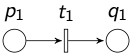


Intermediate Graph

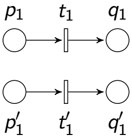


Transformation rule

Left-hand Side

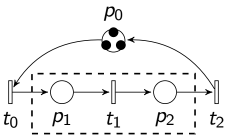


Right-hand Side

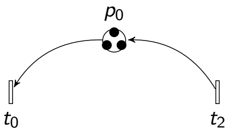


Rule application

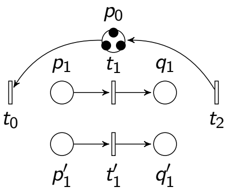
Initial Configuration



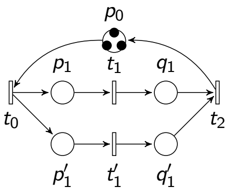
Context Graph



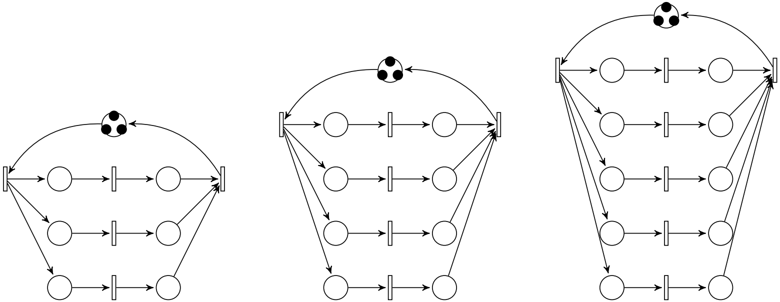
Intermediate Graph



Obtained Configuration



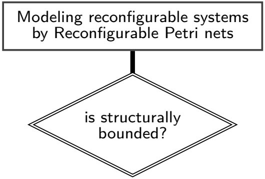
Successive applications of a rewriting rule.

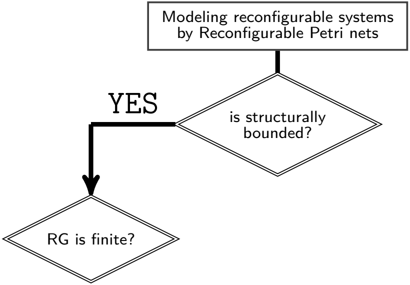


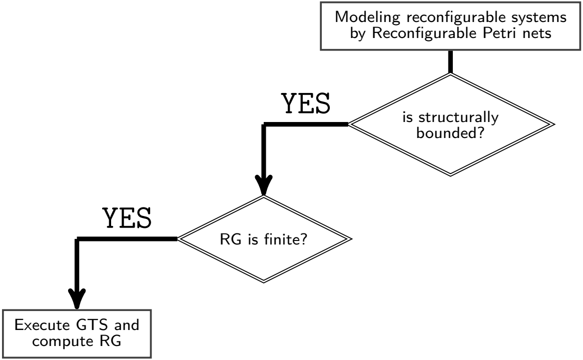
(a) 2nd application. (b) 3rd application. (c) 4th application........

Research Question:
A trade-off between modeling and verification level must be found.

Modeling reconfigurable systems by Reconfigurable Petri nets


Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?


Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?


Modeling reconfigurable systems by Reconfigurable Petri nets

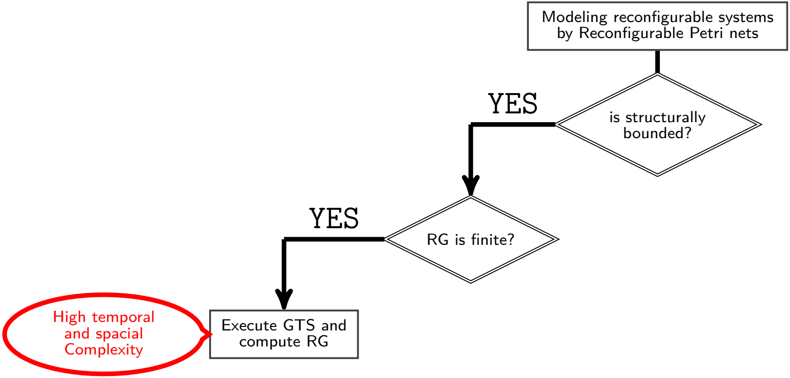
is structurally bounded?

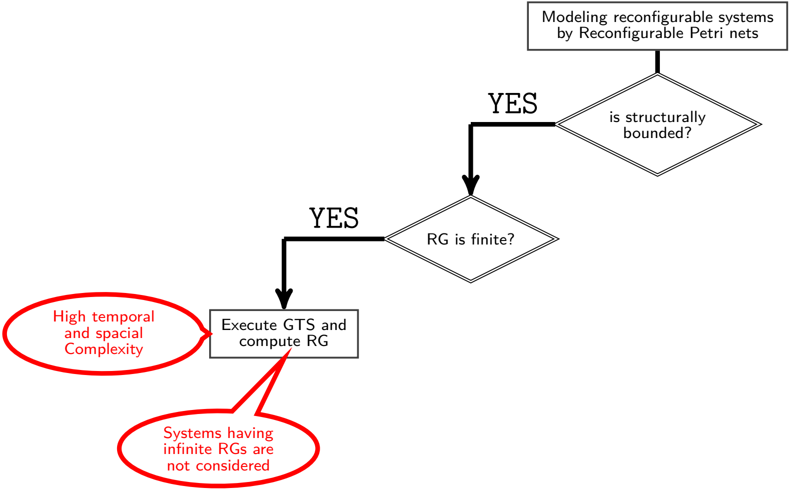
YES

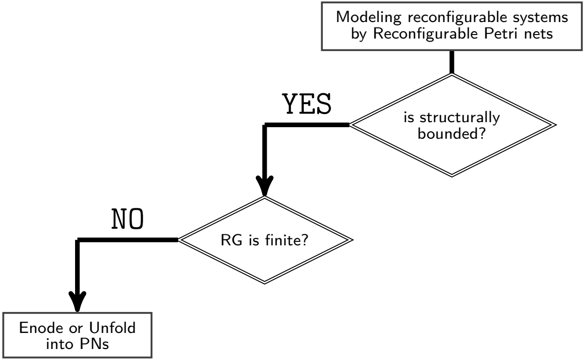
RG is finite?

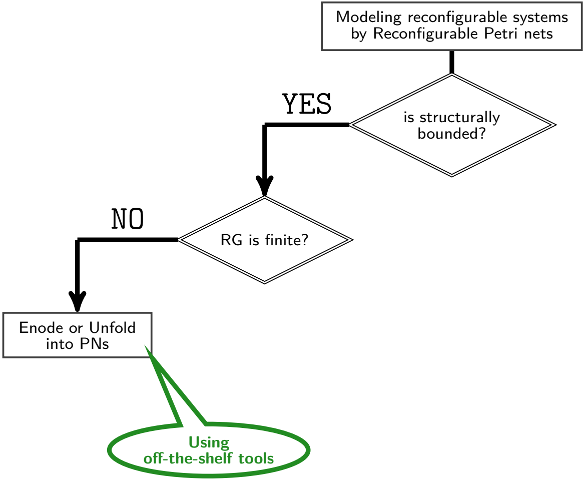
YES

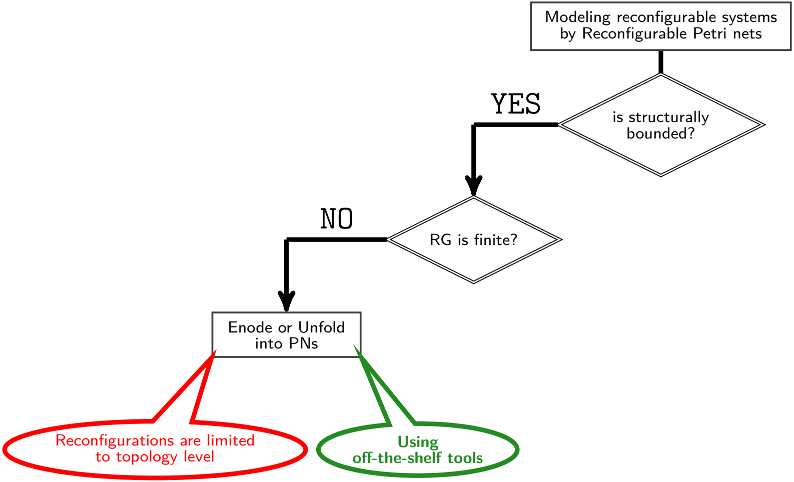
Execute GTS and compute RG

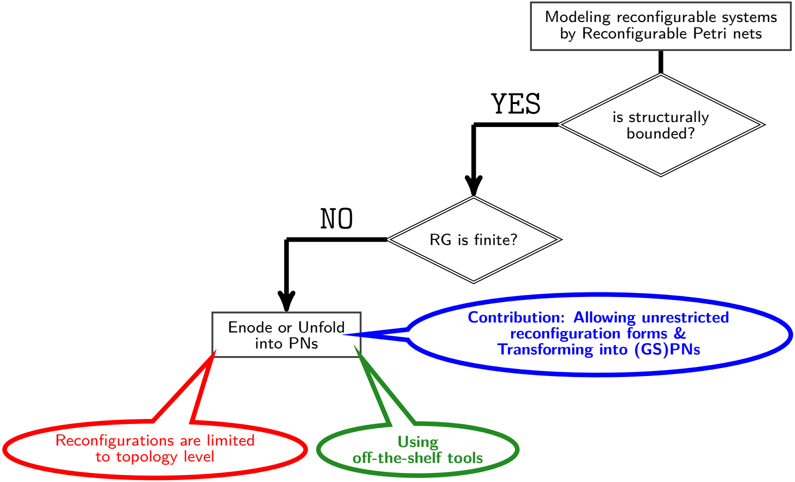

Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?
YES
Execute GTS and compute RG
High temporal and spacial Complexity

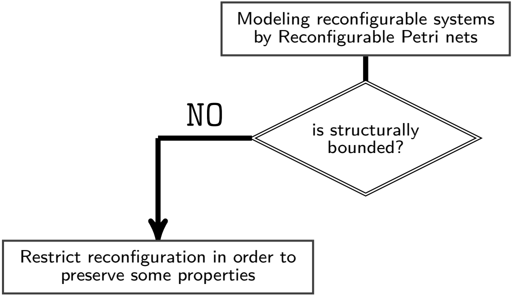

Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?
YES
Execute GTS and compute RG
High temporal and spacial Complexity
Systems having infinite RGs are not considered


Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?
NO
Enode or Unfold into PNs


Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?
NO
Enode or Unfold into PNs
Using off-the-shelf tools


Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?
NO
Enode or Unfold into PNs
Reconfigurations are limited to topology level
Using off-the-shelf tools


Modeling reconfigurable systems by Reconfigurable Petri nets
is structurally bounded?
YES
RG is finite?
NO
Enode or Unfold into PNs
Contribution: Allowing unrestricted reconfiguration forms & Transforming into (GS)PNs
Reconfigurations are limited to topology level
Using off-the-shelf tools

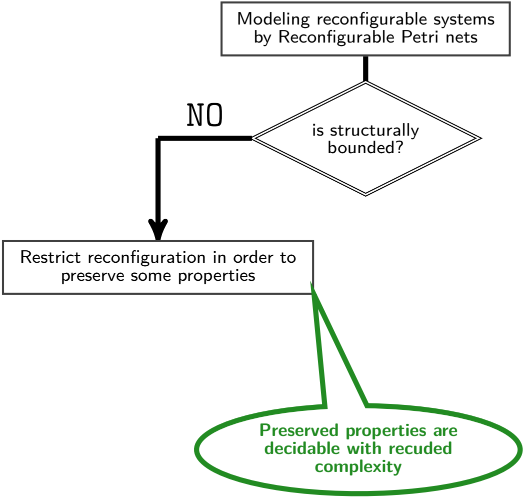


Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

NO

Restrict reconfiguration in order to preserve some properties



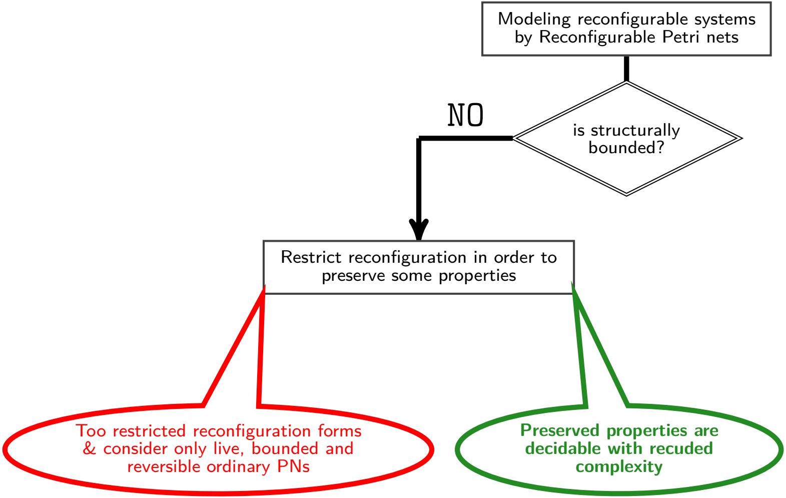
Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

NO

Restrict reconfiguration in order to preserve some properties

Preserved properties are decidable with recuded complexity



Modeling reconfigurable systems by Reconfigurable Petri nets

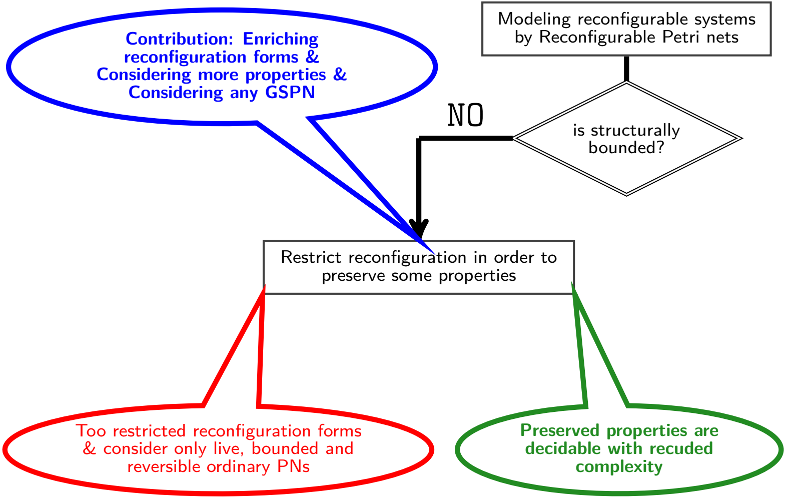
is structurally bounded?

NO

Restrict reconfiguration in order to preserve some properties

Too restricted reconfiguration forms & consider only live, bounded and reversible ordinary PNs

Preserved properties are decidable with recuded complexity



Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

NO

Restrict reconfiguration in order to preserve some properties

Contribution: Enriching reconfiguration forms & Considering more properties & Considering any GSPN

Too restricted reconfiguration forms & consider only live, bounded and reversible ordinary PNs

Preserved properties are decidable with recuded complexity

Contribution: Reconfigurable Generalized Stochastic PNs

RecGSPNs [1] introduce three major advantages:

[1] Samir Tigane *et al.*, "Reconfigurable GSPNs: A modeling formalism of evolvable discrete event systems". *Science of Computer Programming*, 2019.

RecGSPNs [1] introduce three major advantages:

- reconfiguration any GSPN at run-time while preserving several properties,

[1] Samir Tigane *et al.*, "Reconfigurable GSPNs: A modeling formalism of evolvable discrete event systems". *Science of Computer Programming*, 2019.

RecGSPNs [1] introduce three major advantages:

- reconfiguration any GSPN at run-time while preserving several properties,
- a wider range of possible structural changes,

[1] Samir Tigane *et al.*, "Reconfigurable GSPNs: A modeling formalism of evolvable discrete event systems". *Science of Computer Programming*, 2019.

RecGSPNs [1] introduce three major advantages:

- reconfiguration any GSPN at run-time while preserving several properties,
- a wider range of possible structural changes,
- decidability with reduced complexity (infinite graph).

[1] Samir Tigane *et al.*, "Reconfigurable GSPNs: A modeling formalism of evolvable discrete event systems". *Science of Computer Programming*, 2019.

A RecGSPN consists of:

A RecGSPN consists of:

- a GSPN modeling an initial configuration,

A RecGSPN consists of:

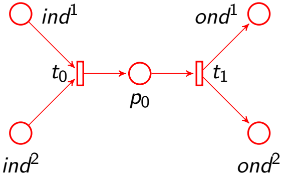
- a GSPN modeling an initial configuration,
- set of rewriting rules.

A RecGSPN consists of:

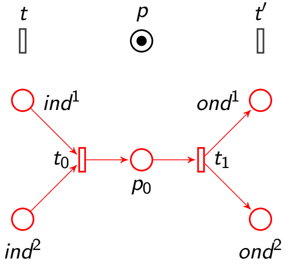
- a GSPN modeling an initial configuration,
- set of rewriting rules.

Both sides of any rule must show a specific behavior.

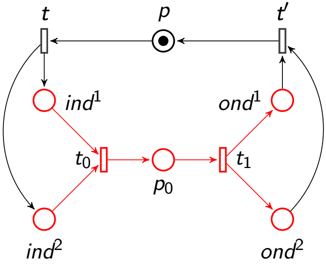
A net N having interface (I_N, O_N) can be used if its container net G is a live, bounded and reversible GSPN:



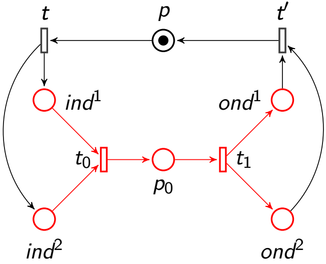
A net N having interface (I_N, O_N) can be used if its container net G is a live, bounded and reversible GSPN:

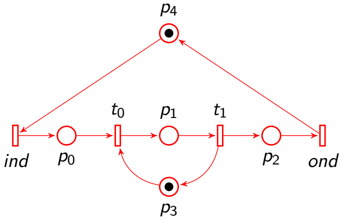


A net N having interface (I_N, O_N) can be used if its container net G is a live, bounded and reversible GSPN:

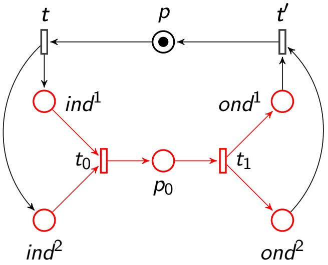


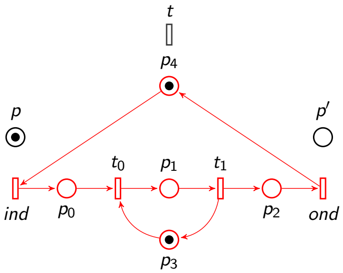
A net N having interface (I_N, O_N) can be used if its container net G is a live, bounded and reversible GSPN:



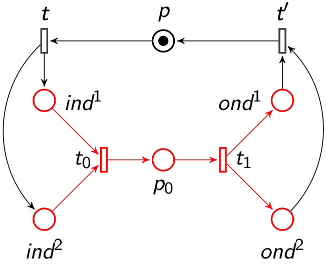


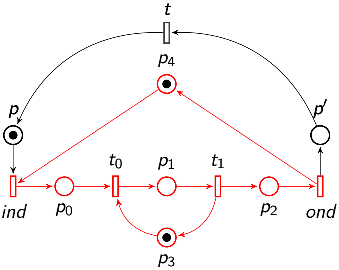
A net N having interface (I_N, O_N) can be used if its container net G is a live, bounded and reversible GSPN:



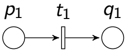


A net N having interface (I_N, O_N) can be used if its container net G is a live, bounded and reversible GSPN:

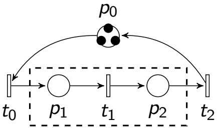




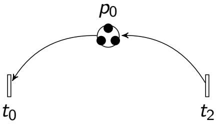
Left-hand Side



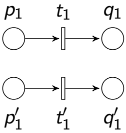
Initial Configuration



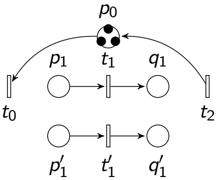
Context Graph



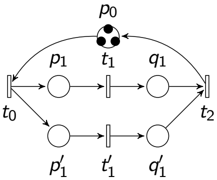
Right-hand Side



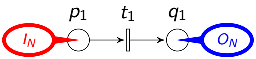
Intermediate Graph



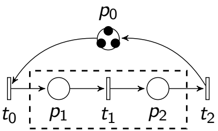
Obtained Configuration



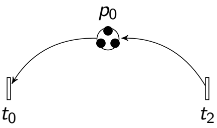
Left-hand Side



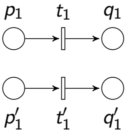
Initial Configuration



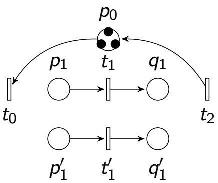
Context Graph



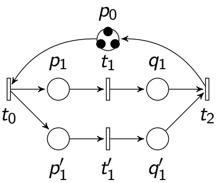
Right-hand Side



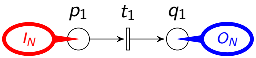
Intermediate Graph



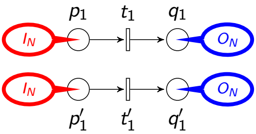
Obtained Configuration



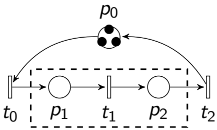
Left-hand Side



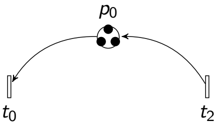
Right-hand Side



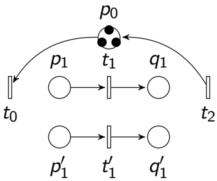
Initial Configuration



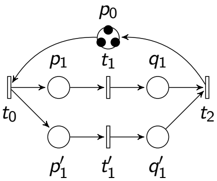
Context Graph

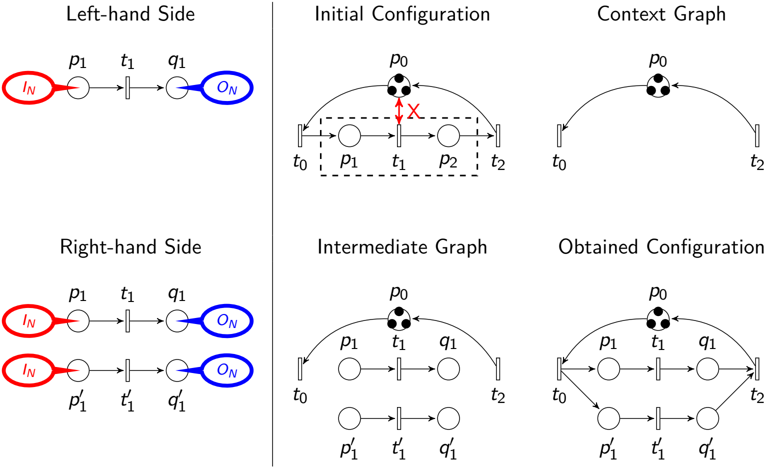


Intermediate Graph



Obtained Configuration





Left-hand Side

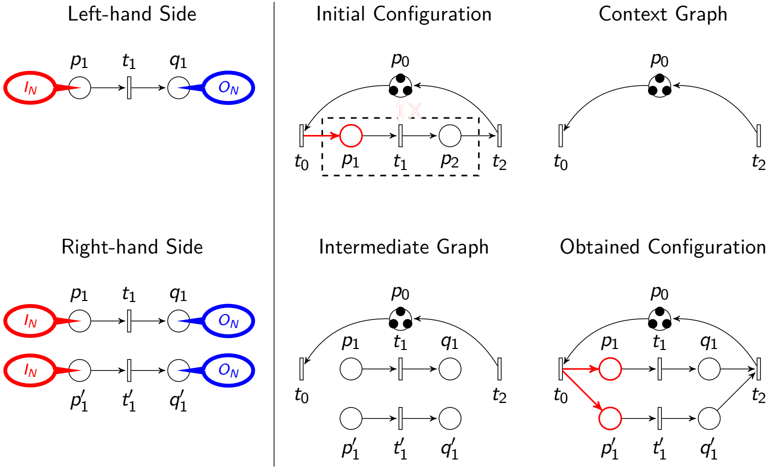
Initial Configuration

Context Graph

Right-hand Side

Intermediate Graph

Obtained Configuration



Left-hand Side

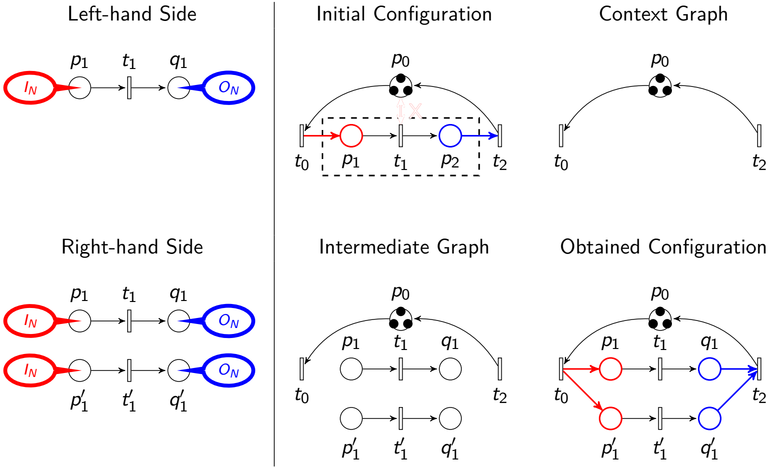
Initial Configuration

Context Graph

Right-hand Side

Intermediate Graph

Obtained Configuration



Left-hand Side

Initial Configuration

Context Graph

Right-hand Side

Intermediate Graph

Obtained Configuration

Reconfiguration preserves

Reconfiguration preserves

- liveness, boundedness, reversibility, home-state, and deadlock-freedom.

Reconfiguration preserves

- liveness, boundedness, reversibility, home-state, and deadlock-freedom.
- LTL_{-next} properties involving only preserved places.

Reconfiguration preserves

- liveness, boundedness, reversibility, home-state, and deadlock-freedom.
- LTL$_{-next}$ properties involving only preserved places.

These properties are decidable with reduced complexity, even if the obtained graph is infinite.

Evaluation

What is the impact of increasing modeling power on decidability?

Formalism	Modeling			Verification		
	US	+/- P/T	GSPNs	QUAL/US	QUAL/BS	QUAN/BS
NRS	✔	✔	✖	✖	✖	✖
RPN	✖	✖	✖	✖	✔	✖
R-TNCES	✖	✖	✖	✖	✔	✔
INRS	✔	✔	✖	✔	✔	✖
Evolving PN	✖	✖	✔	✖	✔	✔
R-SPNs	✖	✖	✖	✖	✔	✔
GSPNs-RT	✖	✖	✔	✖	✔	✔
RecGSPNs	✔	✔	✔	✔	✔	✔

Modeling and verification features: Existing approaches v.s RecGSPNs.

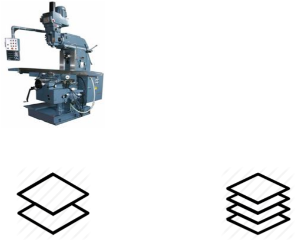
where US, BS, QUAL and QUAN stand for unbounded structure, bounded structure, qualitative, and quantitative.

What is the impact of introducing reconfigurability on spacial and temporal complexity?

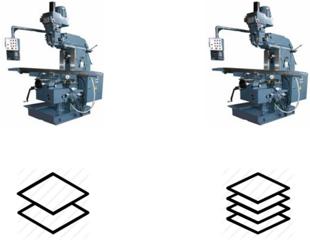
Assume a reconfigurable manufacturing system composed of machine M_1 permanently active and n machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.

Assume a reconfigurable manufacturing system composed of machine M_1 permanently active and *n* machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.

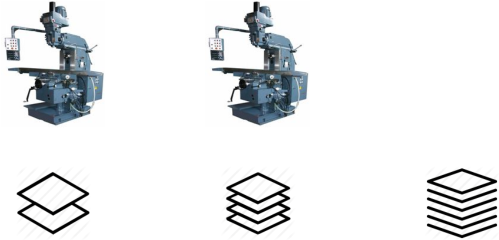
Assume a reconfigurable manufacturing system composed of machine M_1 permanently active and n machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.



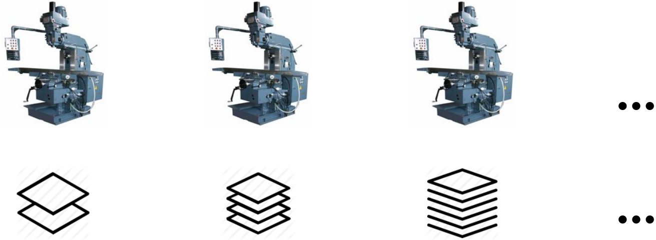
Assume a reconfigurable manufacturing system composed of machine M_1 permanently active and n machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.

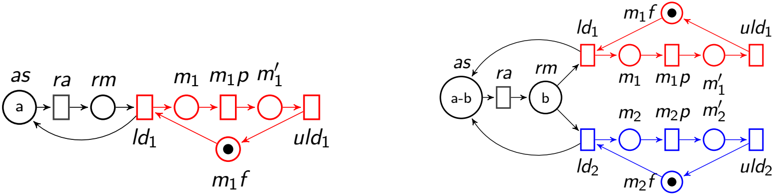


Assume a reconfigurable manufacturing system composed of machine M_1 permanently active and n machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.

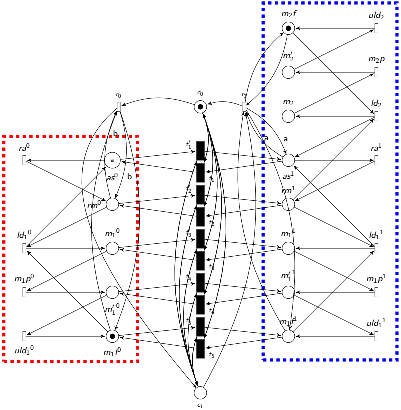


Assume a reconfigurable manufacturing system composed of machine M_1 permanently active and n machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.

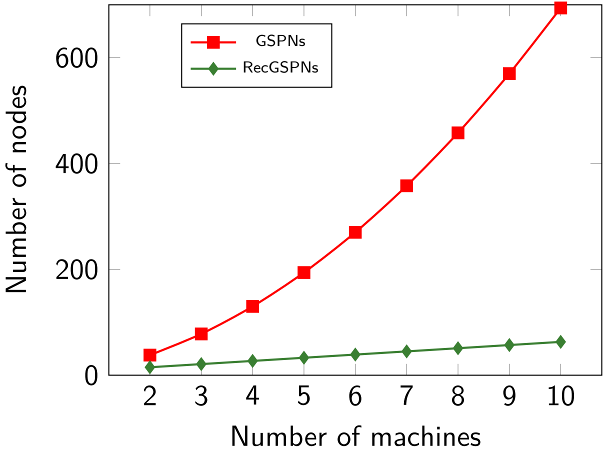




Initial and second configurations.



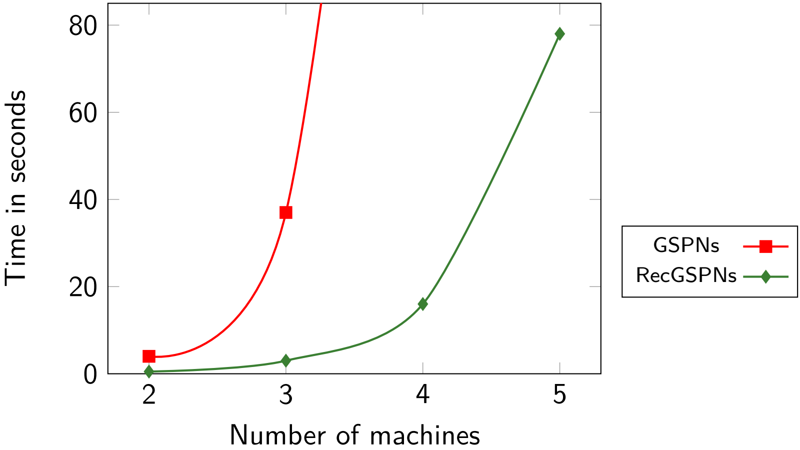
RMS with two machines (based on basic GPSNs).



Factor 1: Model size.

# of machines	RecGSPNs	GSPNs
2	100 states	1963 states
3	220 states	11340 states
4	421 states	State space explosion!
5	743 states	State space explosion!

Factor 2: Semi-Markov chains size according to the number of machines.



Factor 3: Time to compute semi-Markov chains.

Conclusion and Perspectives

Summary

We have proposed a formalism, called RecGSPNs, that allows to preserve several important properties after each reconfiguration.

Summary

We have proposed a formalism, called RecGSPNs, that allows to preserve several important properties after each reconfiguration.

These properties are decidable with reduced complexity even if the structure can be infinite.

Perspectives

We aim to consider:

Perspectives

We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.

Perspectives

We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.
- Reducing the quantitative verification complexity.

Perspectives

We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.
- Reducing the quantitative verification complexity.
- The quantitative properties of structurally unbounded systems.

Selected Papers

- **[1]** S. Tigane, L. Kahloul, S. Benharzallah, S. Baarir, and S. Bourekkache. "Reconfigurable GSPNs: A modeling formalism of evolvable discrete event systems". *Science of Computer Programming*, 183, **2019**.
- **[2]** S. Tigane, L. Kahloul, S. Bourekkache, and S. Baarir. "Extending GSPNs for the modelling, analysis and performance evaluation of dynamic systems". *Int. J. Critical Computer-Based Systems*, 8(1):25–44, **2018**.
- **[3]** S. Tigane, L. Kahloul, and S. Bourekkache. "Reconfigurable stochastic Petri nets for reconfigurable manufacturing systems". *In Proc. 6th Int. Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 383–391. Springer, **2017**.
- **[4]** S. Tigane, L. Kahloul, and S. Bourekkache. "Generalized stochastic Petri nets with rewritable topology". In Proc. of *International Conf. on EDiS*, pages 1–6, **2017**.
- **[5]** S. Tigane, L. Kahloul, and S. Bourekkache. "Reconfigurable stochastic Petri nets: A new formalism for reconfigurable discrete event systems". *In Proc. of ICMIT*, pages 301–308, **2017**.
- **[6]** S. Tigane, L. Kahloul, and S. Bourekkache. "Net rewriting system for GSPN an RMS case study". In Proc. *International Conference on Advanced Aspects of Software Engineering*, pages 38–45. IEEE, **2016**.

Thanks!

Any Questions?