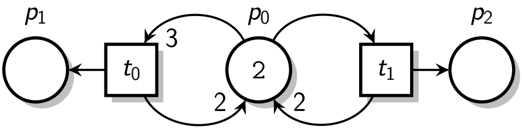
Property Directed Reachability *for Generalized Petri Nets*

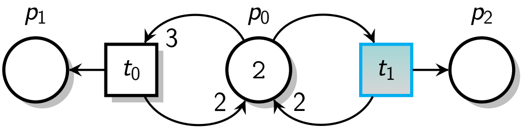
Nicolas Amat, **Silvano Dal Zilio**, **Thomas Hujsa**

LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France

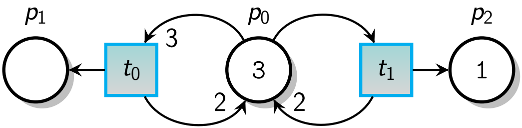
TACAS, April 6 2022



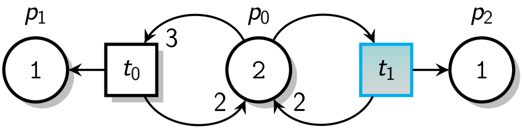
p1
p0
p2
3
t0
2
t1
2
2



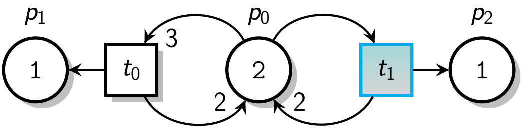
- *Example*: $(2, 0, 0)$



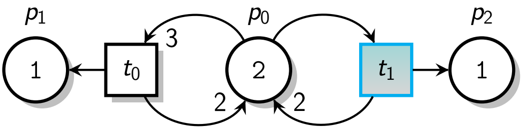
- *Example*: $(2, 0, 0) \xRightarrow{t_1} (3, 0, 1)$



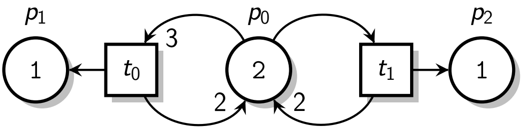
- *Example*: $(2,0,0) \xRightarrow{t_1} (3,0,1) \xRightarrow{t_0} (2,1,1)$



- *Example*: $(2,0,0) \xRightarrow{t_1 . t_0} (2,1,1)$

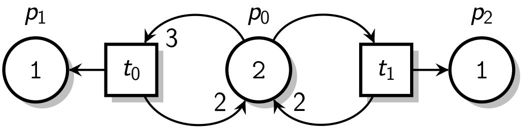


- *Example*: $(2,0,0) \xRightarrow{t_1 . t_0} (2,1,1) \quad \ldots$



- *Example*: $(2,0,0) \xrightarrow{t_1 . t_0} (2,1,1) \ \ \ldots$

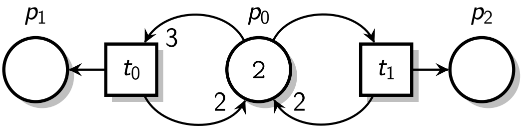
- No constraints on weights of the arcs
- Possibly unbounded



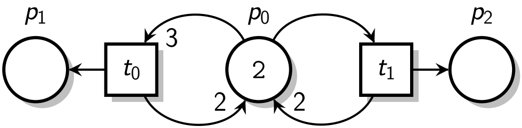
- *Example*: $(2,0,0) \xRightarrow{t_1 . t_0} (2,1,1) \quad \ldots$

- No constraints on weights of the arcs
- Possibly unbounded

- Is $F \triangleq (p_1 \leqslant p_2 \wedge p_0 \geqslant 2)$ an invariant?



$$
\begin{aligned}
\mathrm{ENBL}_{t_0}(\boldsymbol{p}) &\triangleq (p_0 \geqslant 3)\\
\mathrm{ENBL}_{t_1}(\boldsymbol{p}) &\triangleq (p_0 \geqslant 1)\\
\Delta_{t_0}(\boldsymbol{p}, \boldsymbol{p}') &\triangleq (p_0' = p_0 - 1) \wedge (p_1' = p_1 + 1) \wedge (p_2' = p_2)\\
\Delta_{t_1}(\boldsymbol{p}, \boldsymbol{p}') &\triangleq (p_0' = p_0 + 1) \wedge (p_1' = p_1) \wedge (p_2' = p_2 + 1)
\end{aligned}
$$

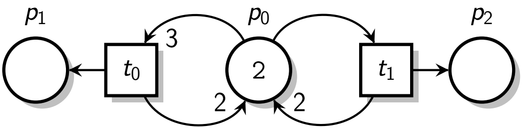


$$
\begin{aligned}
\mathrm{ENBL}_{t_0}(\boldsymbol{p}) &\triangleq (p_0 \geqslant 3)\\
\mathrm{ENBL}_{t_1}(\boldsymbol{p}) &\triangleq (p_0 \geqslant 1)\\
\Delta_{t_0}(\boldsymbol{p}, \boldsymbol{p'}) &\triangleq (p'_0 = p_0 - 1) \wedge (p'_1 = p_1 + 1) \wedge (p'_2 = p_2)\\
\Delta_{t_1}(\boldsymbol{p}, \boldsymbol{p'}) &\triangleq (p'_0 = p_0 + 1) \wedge (p'_1 = p_1) \wedge (p'_2 = p_2 + 1)
\end{aligned}
$$

From this, we can construct the transition relation $\mathrm{T}(\boldsymbol{p}, \boldsymbol{p'})$

Reachable predicate: satisfied by at least one reachable marking

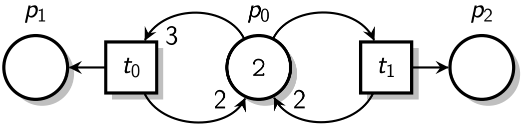
Reachable predicate: satisfied by at least one reachable marking



$$\boldsymbol{F \triangleq (p_2 \geqslant 5)}$$

Invariant predicate: satisfied by all the reachable markings
(its negation is non-reachable)

Invariant predicate: satisfied by all the reachable markings
(its negation is non-reachable)



$$F \triangleq (p_1 \leqslant p_2 \land p_0 \geqslant 2)$$

- Verification of concurrent systems (biological, business processes, ...)
- Verification of software systems

- Verification of concurrent systems
  (biological, business processes, ...)
- Verification of software systems
- Analysis of infinite state systems

- Verification of concurrent systems
  (biological, business processes, ...)
- Verification of software systems
- Analysis of infinite state systems
- Timely subject [Blondin et al. '2021] [Dixon et al. '2020]

- Verification of concurrent systems
  (biological, business processes, ...)
- Verification of software systems
- Analysis of infinite state systems
- Timely subject [Blondin et al. '2021] [Dixon et al. '2020]
- Category of the Model Checking Contest

- Theoretical interest:
  - Equivalent to Vector Addition Systems with States (VASS)
  - Difficult (Ackermann-complete) [Czerwiński et al. '2020]
  - Decidable [Mayr '1981 – Kosaraju '1982],
    but still no complete and efficient method

Related Work

Introduction

- Theoretical interest:
  - Equivalent to Vector Addition Systems with States (VASS)
  - Difficult (Ackermann-complete) [Czerwiński et al. '2020]
  - Decidable [Mayr '1981 – Kosaraju '1982],
    but still no complete and efficient method

- Many tools:
  - ITS-Tools
  - LoLA
  - Tapaal
  - KReach
  - FastForward
  - ...

- Theoretical interest:
  - Equivalent to Vector Addition Systems with States (VASS)
  - Difficult (Ackermann-complete) [Czerwiński et al. '2020]
  - Decidable [Mayr '1981 – Kosaraju '1982],
    but still no complete and efficient method

- Many tools:
  - ITS-Tools
  - LoLA
  - Tapaal
  - KReach
  - FastForward
  - ...

- But efficient methods are missing for (non-coverability) invariant properties on unbounded nets

- Adaptation of PDR for coverability as a testbed for Polyhedral Reductions [Amat et. al '2021]

- Adaptation of PDR for coverability as a testbed for Polyhedral Reductions [Amat et. al '2021]

- Construction of a benchmark composed of some (small) complex nets (out of reach of tools)

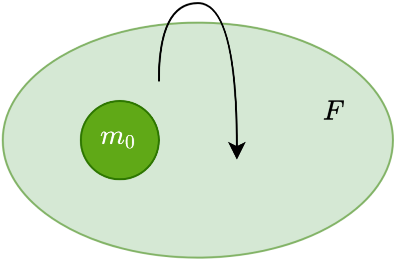
- Adaptation of PDR for coverability as a testbed for Polyhedral Reductions [Amat et. al '2021]
- Construction of a benchmark composed of some (small) complex nets (out of reach of tools)
- Extension to reachability formulas (MCC-like)

- Adaptation of PDR for coverability as a testbed for Polyhedral Reductions [Amat et. al '2021]
- Construction of a benchmark composed of some (small) complex nets (out of reach of tools)
- Extension to reachability formulas (MCC-like)
- Certificate of invariance

Definition (Inductive Predicate)

A linear predicate F is inductive if:

- $m_0 \models F$
- for all m s.t. $m \models F$ we have $m \to m'$ entails $m' \models F$



"There exist checkable certificates of non-reachability in the Presburger arithmetic" [Leroux, 2009]

Definition (Certificate of Invariance)

A predicate R is a Certificate of Invariance (CI) for F if:

- R inductive
- R entails F: $R(\boldsymbol{p}) \wedge \neg F(\boldsymbol{p})$ unsatisfiable

PDR Algorithm

- Also known as *IC3: Incremental Construction of Inductive Clauses for Indubitable Correctness* [Bradley, 2006]
- Symbolic model checking procedure
- Combination of induction, over-approximation, SMT solving

- Also known as *IC3: Incremental Construction of Inductive Clauses for Indubitable Correctness* [Bradley, 2006]

- Symbolic model checking procedure

- Combination of induction, over-approximation, SMT solving

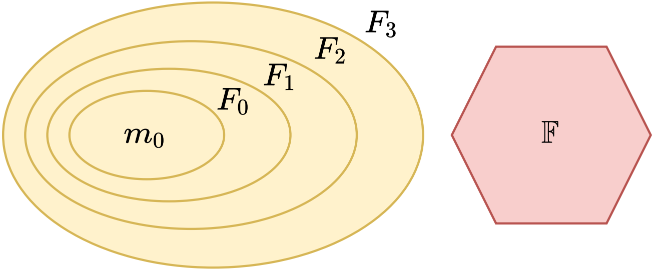
We define:

- $\mathbb{P}$, the invariant that we want to prove on a net (N, m_0)
- $\mathbb{F} = \neg\mathbb{P}$ as the set of feared events (DNF)

Definition

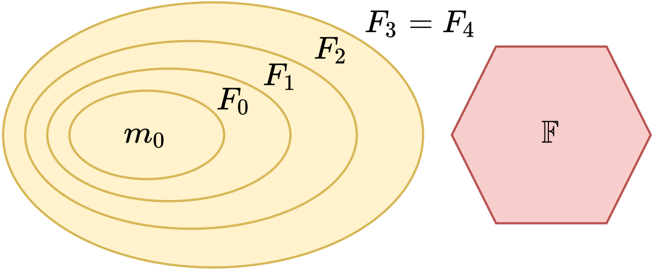
A sequence of formula $F_0, F_1, F_2, \ldots$ such that

1. monotonic: $F_i \Rightarrow F_{i+1}$
2. contains initial state: $F_0 = m_0$
3. does not contain feared state $F_i(\boldsymbol{p}) \wedge \mathbb{F}(\boldsymbol{p})$ unsatisfiable
4. consecution: $F_i(\boldsymbol{p}) \wedge \mathrm{T}(\boldsymbol{p}, \boldsymbol{p'}) \wedge \neg F_{i+1}(\boldsymbol{p'})$ unsatisfiable



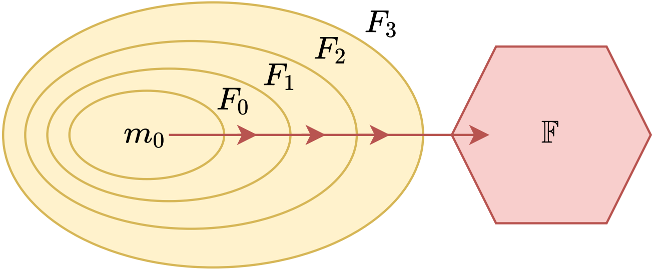
Stop when:

- $F_i = F_{i+1}$: F_i is a certificate of invariance of predicate $\mathbb{P}$
- or, counterexample



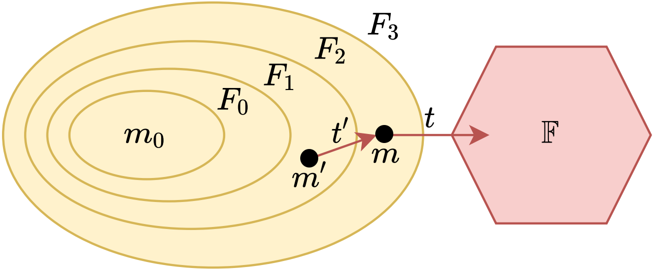
Stop when:

- $F_i = F_{i+1}$: F_i is a certificate of invariance of predicate $\mathbb{P}$
- or, counterexample



We want to generalize scenario such that $m \overset{\sigma}{\Longrightarrow} m_f$ and $m_f \models \mathbb{F}$.

- must be a cube (conjunction),

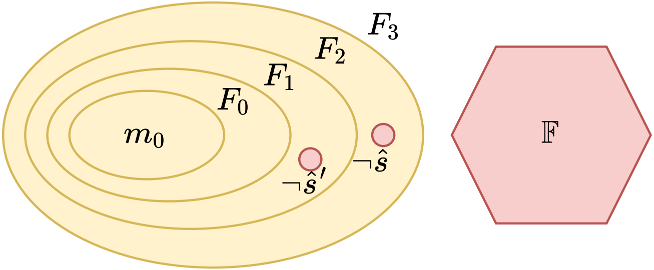


Proof Obligation

PDR Algorithm

We want to generalize scenario such that $m \xrightarrow{\sigma} m_f$ and $m_f \models \mathbb{F}$.

- must be a cube (conjunction),
- assert its negation to block states
- in practice, block the *Minimal Inductive Clause*



Witnesses Generalization

Assume we have a witness scenario (m_1, σ, F), i.e., there exists m_1' such that $m_1 \overset{\sigma}{\Longrightarrow} m_1'$ and $m_1' \models F$ (with F a cube of $\mathbb{F}$)

We have three possible generalizations of the trio (m_1, σ, F)

- **Monotonicity of Petri nets**:
  if $m_1 \overset{\sigma}{\Rightarrow} m_1'$ then for all $m_2 \geqslant m_1$ we have $m_2 \overset{\sigma}{\Rightarrow} m_1' + (m_2 - m_1)$

- **Monotonicity of Petri nets**:
  if $m_1 \overset{\sigma}{\Rightarrow} m'_1$ then for all $m_2 \geqslant m_1$ we have $m_2 \overset{\sigma}{\Rightarrow} m'_1 + (m_2 - m_1)$

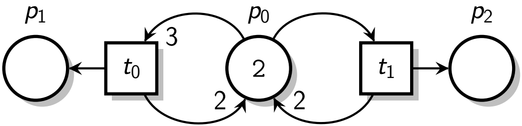
- **Monotonic feared states predicate**:
  if $m'_1 \models F$ then for all $m'_2 \geqslant m'_1$ we have $m'_2 \models F$

- **Monotonicity of Petri nets**:
  if $m_1 \overset{\sigma}{\Rightarrow} m_1'$ then for all $m_2 \geqslant m_1$ we have $m_2 \overset{\sigma}{\Rightarrow} m_1' + (m_2 - m_1)$
- **Monotonic feared states predicate**:
  if $m_1' \models F$ then for all $m_2' \geqslant m_1'$ we have $m_2' \models F$
- **Generalization** of (m_1, σ, F): $(\boldsymbol{p} \geqslant m)$

- **Monotonicity of Petri nets**:
  if $m_1 \overset{\sigma}{\Rightarrow} m'_1$ then for all $m_2 \geqslant m_1$ we have $m_2 \overset{\sigma}{\Rightarrow} m'_1 + (m_2 - m_1)$
- **Monotonic feared states predicate**:
  if $m'_1 \models F$ then for all $m'_2 \geqslant m'_1$ we have $m'_2 \models F$
- **Generalization** of (m_1, σ, F): $(\boldsymbol{p} \geqslant m)$

Lemma (G1)

If property F is monotonic and $m_2 \models (\boldsymbol{p} \geqslant m)$ then (m_2, σ, F) is a witness scenario.

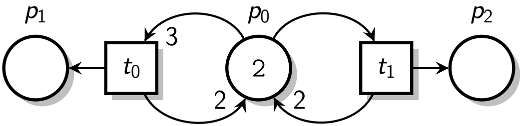


PGCD net, with $\mathbb{F} \triangleq p_1 \geqslant 2$

Scenario: $(3, 1, 0) \xrightarrow{t_0} (2, 2, 0)$ where $(2, 2, 0) \models \mathbb{F}$

(G1) State-based: Concrete Example

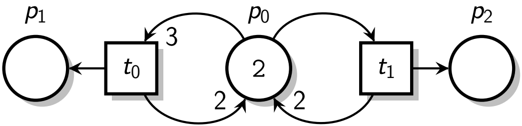
Witness Generalization



PGCD net, with $\mathbb{F} \triangleq p_1 \geqslant 2$

Scenario: $(3, 1, 0) \xRightarrow{t_0} (2, 2, 0)$ where $(2, 2, 0) \models \mathbb{F}$

- **Generalization**: $p_0 \geqslant 3 \land p_1 \geqslant 1$



PGCD net, with $\mathbb{F} \triangleq p_1 \geqslant 2$

Scenario: $(3, 1, 0) \xRightarrow{t_0} (2, 2, 0)$ where $(2, 2, 0) \models \mathbb{F}$

- **Generalization**: $p_0 \geqslant 3 \wedge p_1 \geqslant 1$
- **Learn clause**: $p_0 < 3 \vee p_1 < 1$

But: Only suitable for monotonic predicates!

This known as the *coverability problem*

Given a sequence of transitions σ we define:

Given a sequence of transitions σ we define:

- *Displacement* $\Delta(\sigma)$
  - $\Delta(t) = \mathbf{post}(t) - \mathbf{pre}(t)$
  - $\Delta(t.\sigma') = \Delta(t) + \Delta(\sigma')$
- *Hurdle* $H(\sigma)$ [Hack, 1976]
  - $H(t) = \mathbf{pre}(t)$
  - $H(\sigma_1.\sigma_2) = \max(H(\sigma_1), H(\sigma_2) - \Delta(\sigma_1))$

Given a sequence of transitions σ we define:

- *Displacement* $\Delta(\sigma)$
  - $\Delta(t) = \mathbf{post}(t) - \mathbf{pre}(t)$
  - $\Delta(t.\sigma') = \Delta(t) + \Delta(\sigma')$

- *Hurdle* $H(\sigma)$ [Hack, 1976]
  - $H(t) = \mathbf{pre}(t)$
  - $H(\sigma_1.\sigma_2) = \max(H(\sigma_1), H(\sigma_2) - \Delta(\sigma_1))$

Hence, $m \overset{\sigma}{\Longrightarrow} m'$ if and only if:

1. the sequence σ is enabled at m: $m \geqslant H(\sigma)$
2. and $m' = m + \Delta(\sigma)$

- Generalize sequences instead of states

- Generalize sequences instead of states
- **Generalization** of (m_1, σ, F): $(\boldsymbol{p} \geqslant H(\sigma) \wedge F(\boldsymbol{p} + \Delta(\sigma)))$

- Generalize sequences instead of states
- **Generalization** of (m_1, σ, F): $(\boldsymbol{p} \geqslant H(\sigma) \wedge F(\boldsymbol{p} + \Delta(\sigma)))$

Lemma (G2)

If $m_2 \models \boldsymbol{p} \geqslant H(\sigma) \wedge F(\boldsymbol{p} + \Delta(\sigma))$ then (m_2, σ, F) is a witness scenario.

We define the Hurdle of a saturated sequence of transitions σ^{k+1}:

$$H(\sigma^{k+1}) = \max\left(H(\sigma), H(\sigma) - k \cdot \Delta(\sigma)\right) = H(\sigma) + k \cdot (-\Delta(\sigma))^{+}$$

We define the Hurdle of a saturated sequence of transitions σ^{k+1}:

$$H(\sigma^{k+1}) = \max\left(H(\sigma), H(\sigma) - k \cdot \Delta(\sigma)\right) = H(\sigma) + k \cdot (-\Delta(\sigma))^{+}$$

And so, $m \overset{\sigma}{\Rightarrow} \overset{\sigma^k}{\Longrightarrow} m'$ if and only if

1. $m \geqslant H(\sigma) + k \cdot \max(\mathbf{0}, -\Delta(\sigma))$
2. $m' = m + (k+1) \cdot \Delta(\sigma)$

Lemma (G3)

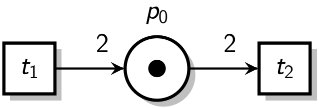
Assume a, b are mappings of $\mathbb{N}^P$ s.t. $a = H(\sigma)$ and $b = (-\Delta(\sigma))^+$

$$m_2 \models \exists k. \left(\begin{array}{l} [\boldsymbol{p} \geqslant a + k \cdot b)] \\ \wedge F(\boldsymbol{p} + (k+1) \cdot \Delta(\sigma)) \end{array} \right)$$

implies $\exists k$ such that (m_2, σ^{k+1}, F) is a witness scenario.

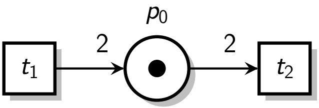
(G3) Saturated Transition-based: Concrete Example

Witness Generalization



Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

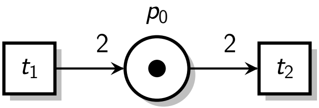
Scenario: $(2) \stackrel{t_2}{\Longrightarrow} (0)$



Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

Scenario: $(2) \xRightarrow{t_2} (0)$

- $H(t_2^{k+1}) = (2.(k+1))$ and $\Delta(t_2^{k+1}) = (-2.(k+1))$



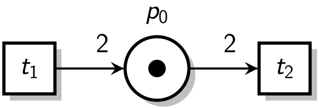
Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

Scenario: $(2) \xRightarrow{t_2} (0)$

- $H(t_2^{k+1}) = (2.(k+1))$ and $\Delta(t_2^{k+1}) = (-2.(k+1))$
- **Generalization**: $\exists k.\ ((p_0 \geqslant 2.(k+1)) \wedge (p_0 - 2.(k+1) \geqslant 1))$
  $\equiv \exists k.\ (p_0 = 2.(k+1))$

(G3) Saturated Transition-based: Concrete Example

Witness Generalization



Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

Scenario: $(2) \xRightarrow{t_2} (0)$

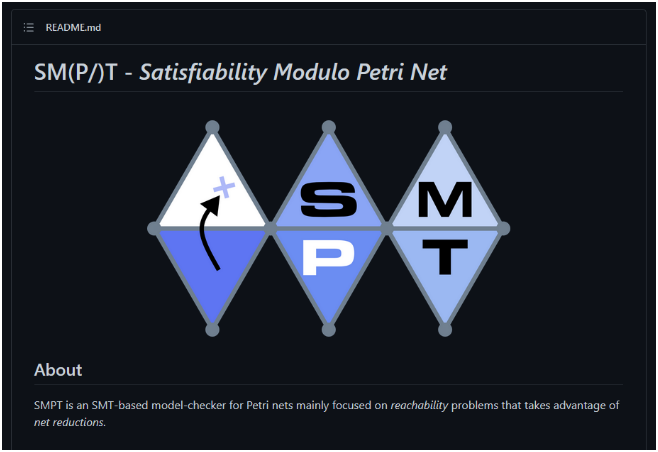
- $H(t_2^{k+1}) = (2.(k+1))$ and $\Delta(t_2^{k+1}) = (-2.(k+1))$
- **Generalization**: $\exists k.\ ((p_0 \geqslant 2.(k+1)) \land (p_0 - 2.(k+1) \geqslant 1))$
  $\equiv \exists k.\ (p_0 = 2.(k+1))$
- **Learn clause**: $\forall k.(p_0 \neq 2.(k+1))$

Experimental Results

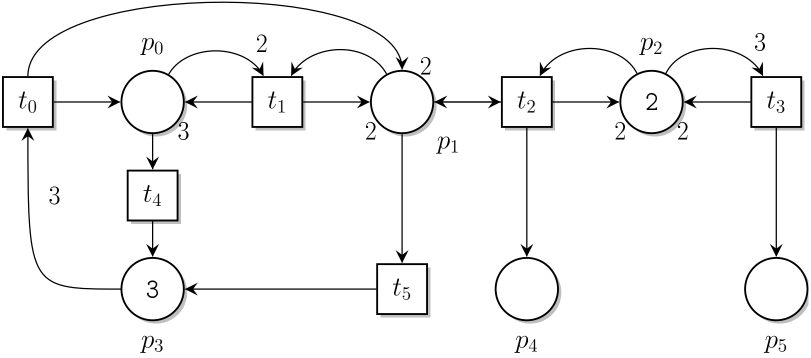
https://github.com/nicolasAmat/smpt

SMPT: Prototpype Model Checker

Experimental Results



`https://github.com/nicolasAmat/smpt`



Murphy net, with $\mathbb{P} \triangleq (p_1 \leqslant 2 \wedge p_4 \geqslant p_5)$

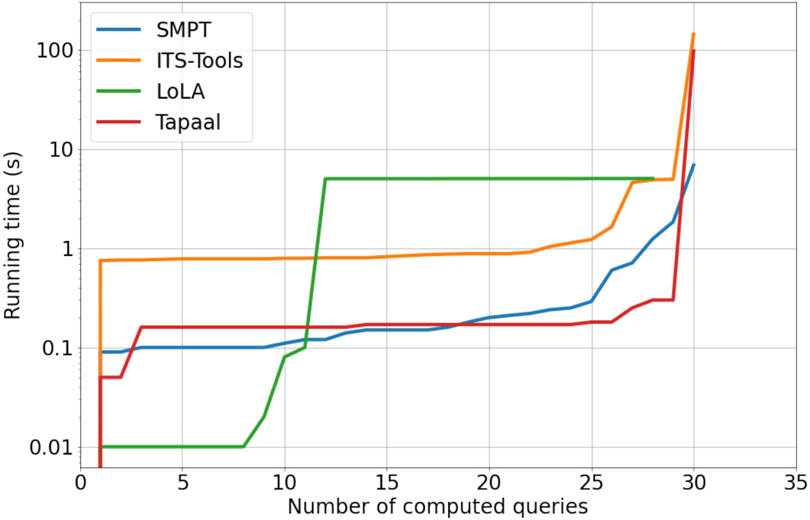
ITS-Tools, LoLA, Tapaal: *k*-induction, state equation, walker, trace abstract refinement, etc.

Instance	SMPT	ITS-Tools	LoLA	Tapaal
Murphy	0.75 *	TLE	TLE	TLE
PGCD	0.11 *	139.08	TLE	TLE
CryptoMiner	0.19 *	5.92	TLE	0.18
Parity	0.40 *	3.36	0.01	4.16
Process	83.39	TLE	0.03	0.18

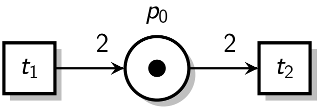
*: use of saturation
TLE: Time Limit Exceeded (1 h)

Comparison on Performance

Experimental Results



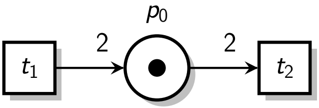
Tool Certification



Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

Certificate of Invariance

Tool Certification

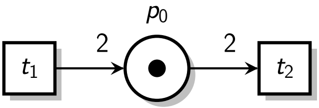


Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

```
################################
[PDR] Certificate of invariance
# (not (p0 < 1))
# (forall (k1) ((p0 < (2 + (k1 * 2))) or ((p0 + (-2 * (k1 + 1))) >= 1)
################################
[PDR] Certificate checking
# UNSAT(I /\ -Proof): True
# UNSAT(R /\ Proof): True
# UNSAT(Proof /\ T /\ -Proof'): True
################################
FORMULA Parity-Inv TRUE TIME
```

Certificate of Invariance

Tool Certification

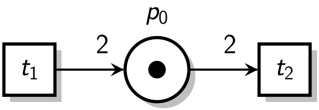


Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

```
[PDR] Certificate of invariance
# (not (p0 < 1))
# (forall (k1) ((p0 < (2 + (k1 * 2))) or ((p0 + (-2 * (k1 + 1))) >= 1)
```

$\mathbb{C} \equiv (p_0 \geqslant 1) \wedge \forall k.((p_0 < 2\,k + 2) \vee (p_0 \geqslant 2\,k + 3))$

- equivalent to $(p_0 \geqslant 1) \wedge \forall k.(p_0 \neq 2.(k + 1)$
- meaning the marking of p_0 is odd

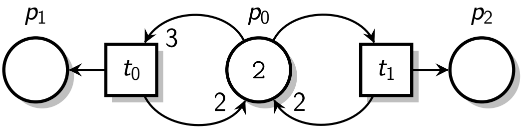


Parity, with invariant $\mathbb{P} = (p_0 \geqslant 1)$

```
[PDR] Certificate checking
# UNSAT(I /\ -Proof): True
# UNSAT(R /\ Proof): True
# UNSAT(Proof /\ T /\ -Proof'): True
```

Not need to trust our tool: it provide a checkable proof!

Certificate of Invariance

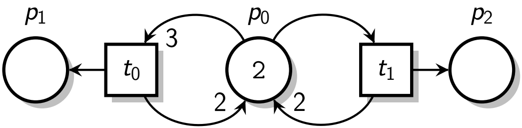


PGCD, with invariant $\mathbb{P} = (p_1 \leqslant p_2)$

```
[PDR] Certificate of invariance
# (not (p1 > p2))
# (forall (k1) ((p0 < (3 + (k1 * 1))) or ((p1 + (1 * (k1 + 1))) <= p2))
```

Certificate of Invariance

Tool Certification



PGCD, with invariant $\mathbb{P} = (p_1 \leqslant p_2)$

```
[PDR] Certificate of invariance
# (not (p1 > p2))
# (forall (k1) ((p0 < (3 + (k1 * 1))) or ((p1 + (1 * (k1 + 1))) <= p2))
```

$$\mathbb{C} \equiv (p_1 \leqslant p_2) \wedge \forall k.((p_0 < k + 3) \vee (p_2 - p_1 \geqslant k + 1))$$

- saturation "learned" the invariant $p_0 + p_1 = p_2 + 2$
- use it to strengthen property $\mathbb{P}$ into an inductive invariant (Property Directed)

Conclusion and Perspectives

Conclusion

- We propose a method that works as well on bounded as on unbounded nets
- Behaves well when the invariant is true
- Works with “genuine” reachability properties
- Provide *certificate of invariance*

Perspectives

Is our method complete?

Is our method complete?

- Complete for coverability properties

Perspectives

Is our method complete?

- Complete for coverability properties
- Incomplete without the saturation

Perspectives

Is our method complete?

- Complete for coverability properties
- Incomplete without the saturation
- Open problem

Perspectives

Is our method complete?

- Complete for coverability properties
- Incomplete without the saturation
- Open problem
- If a proof exists, it would be complicated (cf. Kosaraju's proof)

Thank you for your attention!

Any questions?