A Polyhedral Framework for Reachability Problems in Petri Nets

Un cadre polyédrique pour les problèmes d'accessibilité dans les réseaux de Petri

Nicolas Amat

François Vernadat, Didier Le Botlan, Silvano Dal Zilio

December 4, 2023

LAAS CNRS

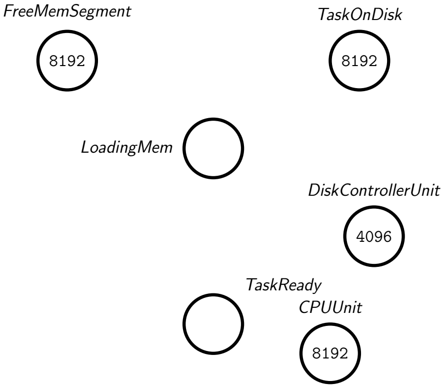
General context

- Verification of **concurrent systems**
- **Model checking** [Emerson and Clarke, 80] [Queille and Sifakis, 82]

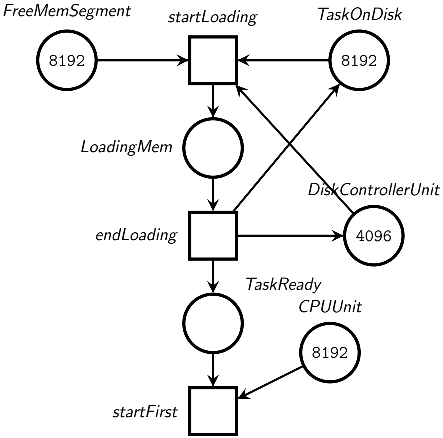
Does an abstract model satisfy a formal specification?

The SmallOperatingSystem example

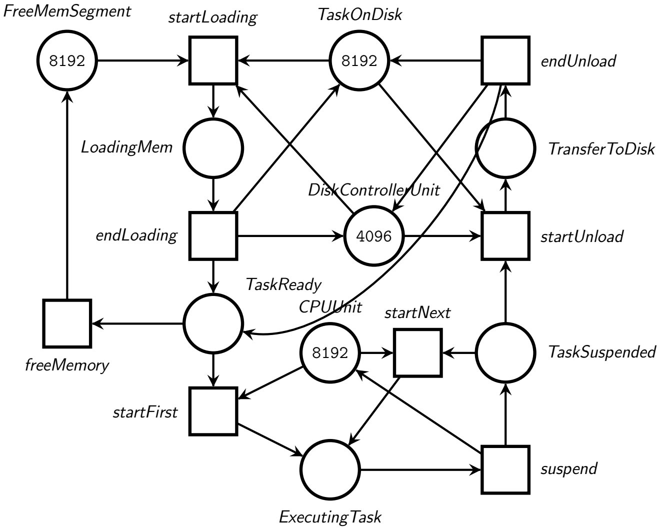
The SmallOperatingSystem example



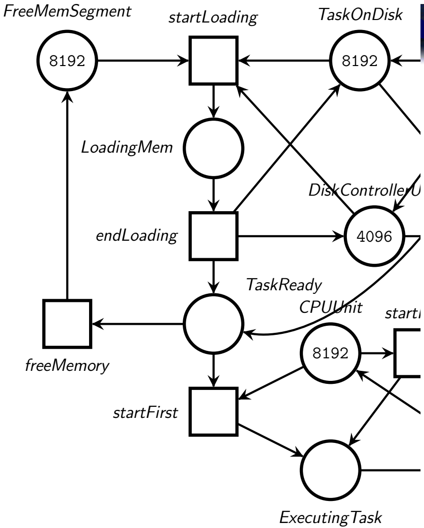
The SmallOperatingSystem example

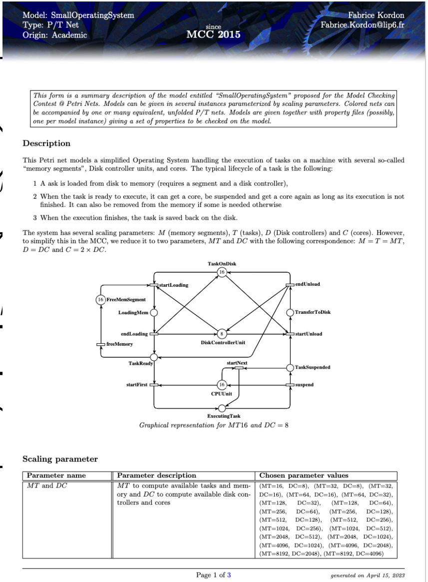


The SmallOperatingSystem example



The SmallOperatingSystem example





Model: SmallOperatingSystem
Type: P/T Net
Origin: Academic

since
MCC 2015

Fabrice Kordon
Fabrice.Kordon@lip6.fr

This form is a summary description of the model entitled "SmallOperatingSystem" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

Description

This Petri net models a simplified Operating System handling the execution of tasks on a machine with several so-called "memory segments", Disk controller units, and cores. The typical lifecycle of a task is the following:

1 A ask is loaded from disk to memory (requires a segment and a disk controller),

2 When the task is ready to execute, it can get a core, be suspended and get a core again as long as its execution is not finished. It can also be removed from the memory if some is needed otherwise

3 When the execution finishes, the task is saved back on the disk.

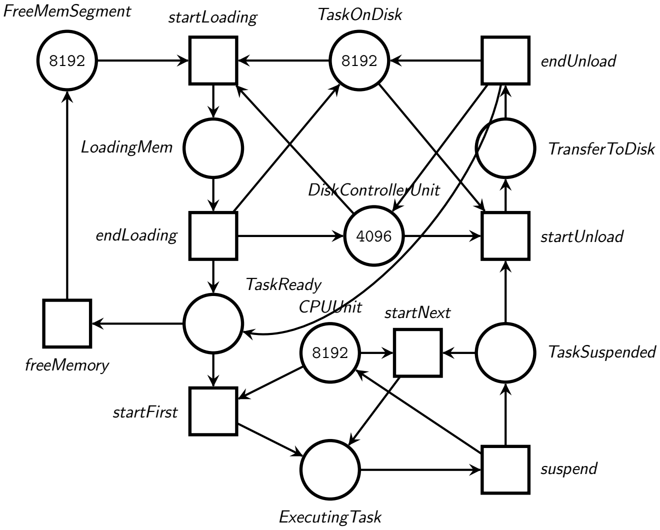
The system has several scaling parameters: M (memory segments), T (tasks), D (Disk controllers) and C (cores). However, to simplify this in the MCC, we reduce it to two parameters, MT and DC with the following correspondence: $M = T = MT$, $D = DC$ and $C = 2 \times DC$.

Graphical representation for $MT16$ and $DC = 8$

Scaling parameter

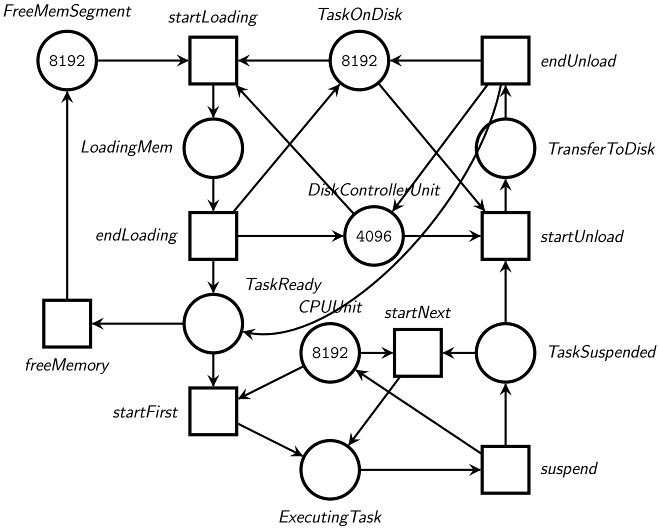
Parameter name	Parameter description	Chosen parameter values
MT and DC	MT to compute available tasks and memory and DC to compute available disk controllers and cores	(MT=16, DC=8), (MT=32, DC=8), (MT=32, DC=16), (MT=64, DC=16), (MT=64, DC=32), (MT=128, DC=32), (MT=128, DC=64), (MT=256, DC=64), (MT=256, DC=128), (MT=512, DC=128), (MT=512, DC=256), (MT=1024, DC=256), (MT=1024, DC=512), (MT=2048, DC=512), (MT=2048, DC=1024), (MT=4096, DC=1024), (MT=4096, DC=2048), (MT=8192, DC=2048), (MT=8192, DC=4096)

The SmallOperatingSystem example



Is "*ExecutingTask* > *TaskOnDisk*" reachable from the initial marking?

The SmallOperatingSystem example



State space $\approx 10^{17}$

Techniques

- **State-space construction**
  - Decision Diagrams
  - Partial Order Reductions, symmetries, etc.
  - Not adapted for reachability problems and cannot handle unbounded nets

Techniques

- **State-space construction**
  - Decision Diagrams
  - Partial Order Reductions, symmetries, etc.
  - Not adapted for reachability problems and cannot handle unbounded nets

- **Portfolio of methods**

- **SMT-based model checking** (thanks to the progress of the solvers)
  - Counter-examples: `BMC`
  - Invariants: k`-induction`, `CEGAR`, `PDR`

Techniques

- **State-space construction**
  - Decision Diagrams
  - Partial Order Reductions, symmetries, etc.
  - Not adapted for reachability problems and cannot handle unbounded nets
- **Portfolio of methods**
- **SMT-based model checking** (thanks to the progress of the solvers)
  - Counter-examples: `BMC`
  - Invariants: k`-induction`, `CEGAR`, `PDR`
- **Optimizations**
  - Structural reductions, slicing, etc.

Techniques

- **State-space construction**
  - Decision Diagrams
  - Partial Order Reductions, symmetries, etc.
  - Not adapted for reachability problems and cannot handle unbounded nets
- **Portfolio of methods**
- **SMT-based model checking** (thanks to the progress of the solvers)
  - Counter-examples: `BMC`
  - Invariants: k`-induction`, `CEGAR`, `PDR`
- **Optimizations**
  - Structural reductions, slicing, etc.

Our approach is complementary!

A polyhedral framework for reachability problems in Petri nets

Petri nets

A strength of Petri net theory is the ability to reuse results from **linear algebra**, and linear programming techniques, to reason on it:

Petri nets

A strength of Petri net theory is the ability to reuse results from **linear algebra**, and linear programming techniques, to reason on it:

- **Potentially reachable markings**, aka the State Equation

$$m = I.\sigma + m_0$$

Petri nets

A strength of Petri net theory is the ability to reuse results from **linear algebra**, and linear programming techniques, to reason on it:

- **Potentially reachable markings**, aka the State Equation

$$m = I.\sigma + m_0$$

- **Place invariants**

$$\sigma^T.I = \mathbf{0}$$

- ...

Petri nets

Some transition t enabled at m when $m \models \mathrm{ENBL}_t(\boldsymbol{p})$:

$$\mathrm{ENBL}_t(\boldsymbol{p}) \triangleq \bigwedge_{i \in 1..n} (p_i \geqslant \mathrm{Pre}(t, p_i))$$

We have $m \to m'$ if and only if $m, m' \models \mathrm{T}(\boldsymbol{p}, \boldsymbol{p'})$:

$$\mathrm{T}(\boldsymbol{p}, \boldsymbol{p'}) \;\triangleq\; \bigvee\nolimits_{t \in T} \mathrm{ENBL}_t(\boldsymbol{p}) \wedge \Delta_t(\boldsymbol{p}, \boldsymbol{p'})$$

where the token displacement is defined as:

$$\Delta_t(\boldsymbol{p}, \boldsymbol{p'}) \;\triangleq\; \bigwedge\nolimits_{i \in 1..n} (p'_i = p_i + \mathrm{Post}(t)(p_i) - \mathrm{Pre}(t)(p_i))$$

Petri nets

Some transition t enabled at m when $m \models \mathrm{ENBL}_t(\boldsymbol{p})$:

$$\mathrm{ENBL}_t(\boldsymbol{p}) \triangleq \bigwedge_{i \in 1..n} (p_i \geqslant \mathrm{Pre}(t, p_i))$$

We have $m \to m'$ if and only if $m, m' \models \mathrm{T}(\boldsymbol{p}, \boldsymbol{p'})$:

$$\mathrm{T}(\boldsymbol{p}, \boldsymbol{p'}) \;\triangleq\; \bigvee_{t \in T} \mathrm{ENBL}_t(\boldsymbol{p}) \wedge \Delta_t(\boldsymbol{p}, \boldsymbol{p'})$$

where the token displacement is defined as:

$$\Delta_t(\boldsymbol{p}, \boldsymbol{p'}) \;\triangleq\; \bigwedge_{i \in 1..n} (p'_i = p_i + \mathrm{Post}(t)(p_i) - \mathrm{Pre}(t)(p_i))$$

In general the relation $m \to^* m'$ cannot be encoded in the Presburger arithmetic

Petri nets

Some transition t enabled at m when $m \models \mathrm{ENBL}_t(\boldsymbol{p})$:

$$\mathrm{ENBL}_t(\boldsymbol{p}) \triangleq \bigwedge_{i \in 1..n} (p_i \geqslant \mathrm{Pre}(t, p_i))$$

We have $m \to m'$ if and only if $m, m' \models \mathrm{T}(\boldsymbol{p}, \boldsymbol{p'})$:

$$\mathrm{T}(\boldsymbol{p}, \boldsymbol{p'}) \triangleq \bigvee\nolimits_{t \in T} \mathrm{ENBL}_t(\boldsymbol{p}) \wedge \Delta_t(\boldsymbol{p}, \boldsymbol{p'})$$

where the token displacement is defined as:

$$\Delta_t(\boldsymbol{p}, \boldsymbol{p'}) \triangleq \bigwedge\nolimits_{i \in 1..n} (p'_i = p_i + \mathrm{Post}(t)(p_i) - \mathrm{Pre}(t)(p_i))$$

In general the relation $m \to^* m'$ cannot be encoded in the Presburger arithmetic

Same formalism for semantics and properties

A polyhedral framework for reachability problems in Petri nets

Reachability properties verification

- F **reachable** if and only if $\exists m \in R(N, m_0)$ such that $m \models F$

Reachability properties verification

- F **reachable** if and only if $\exists m \in R(N, m_0)$ such that $m \models F$
- F **invariant** if and only if $\forall m \in R(N, m_0)$ we have $m \models F$

Reachability properties verification

- F **reachable** if and only if $\exists m \in R(N, m_0)$ such that $m \models F$
- F **invariant** if and only if $\forall m \in R(N, m_0)$ we have $m \models F$

$$\mathrm{EF}\, F \equiv \neg\,(\mathrm{AG}\,\neg F)$$

	$\top$	$\bot$
$\mathrm{EF}\,F$	Witness	Non-reachable
$\mathrm{AG}\,F$	Invariant	Counter-example

Some properties of interest

- **Coverability**: $\mathrm{COVER}(p, k) \;\equiv\; m(p) \geq k$
- **Reachability**: $\mathrm{REACH}(p, k) \;\equiv\; m(p) = k$
- **Quasi-liveness**: $\mathrm{QLIVE}(t) \;\equiv\; \bigwedge_{p \in {}^\bullet t} \mathrm{COVER}(p, \mathbf{pre}(t, p))$
- **Deadlock**: $\mathrm{DEAD} \;\equiv\; \displaystyle\bigwedge_{t \in T} \neg\mathrm{QLIVE}(t)$

Reachability problems

- **Decidable** [Mayr, 1981] [Kosaraju, 1982] [Lambert, 1992]
  ... but still no complete and efficient method.

Reachability problems

- **Decidable** [Mayr, 1981] [Kosaraju, 1982] [Lambert, 1992]
  ... but still no complete and efficient method.
- Difficult (**Ackermann-complete**) [Czerwiński et al., 2022] [Leroux, 2022]

Reachability problems

- **Decidable** [Mayr, 1981] [Kosaraju, 1982] [Lambert, 1992]
  ... but still no complete and efficient method.
- Difficult (**Ackermann-complete**) [Czerwiński et al., 2022] [Leroux, 2022]
- **Many tools**
  - ITS-Tools
  - LoLA
  - TAPAAL
  - KReach
  - FastForward
  - ...

A polyhedral framework for reachability problems in Petri nets

Net reductions [Berthelot, 76]

A reduction is a net transformation which reduces its size such that (for a given set of properties) the reduced net is equivalent to the initial one.

$$(N, m_0) \;\equiv\; (N', m_0')$$

A reduction is characterized by:

- (Graph) transformation
- Application of conditions
- The preserved properties: boundedness; deadlock; quasi-liveness; reachability; ...

Polyhedral reductions

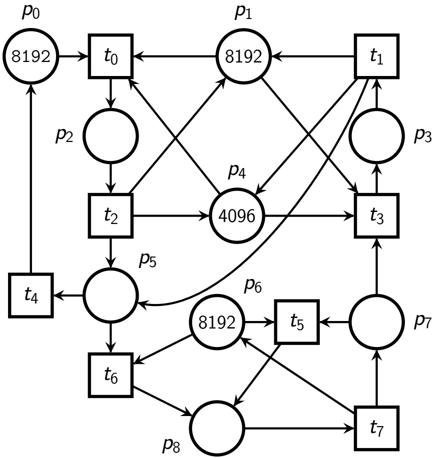
A polyhedral reduction is a net transformation which reduces its size such that we can reconstruct the state space of the initial net from the reduced one.

$$(N, m_0) \equiv_{\mathbf{E}} (N', m'_0)$$

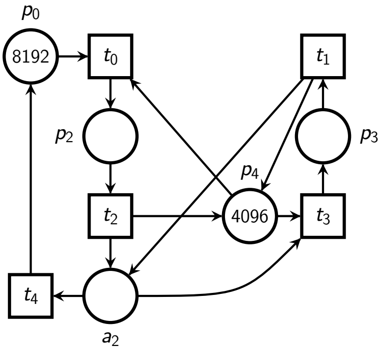
A polyhedral reduction is characterized by:

- A Presburger predicate, **E**, of linear constraints between places.
- (Graph) transformation
- Application of conditions
- The preserved properties: ~~boundedness~~; ~~deadlock~~; ~~quasi-liveness~~; **reachability**; ...

SmallOperatingSystem

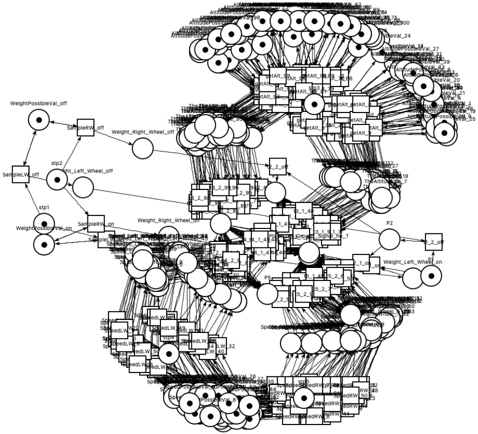


$\equiv_E$

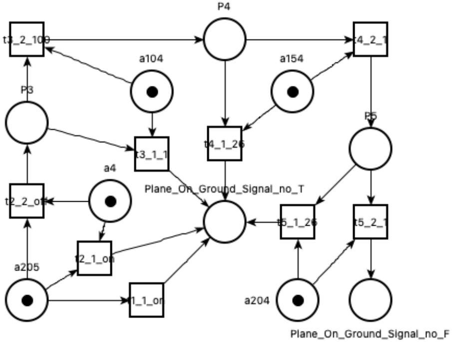


$$E \triangleq \exists a_1. \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_6 \end{cases}$$

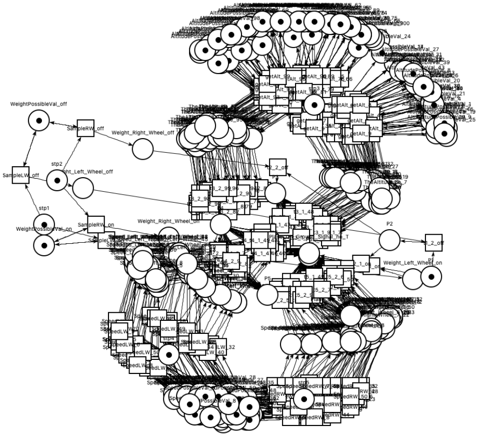
AirplaneLD-PT-0050



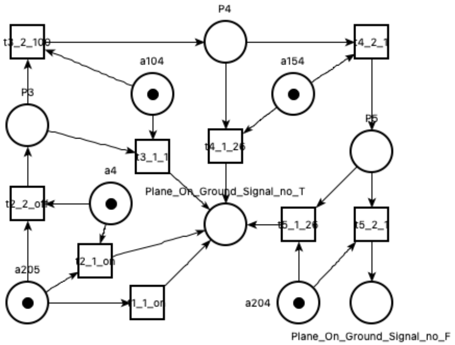
$\equiv_E$



AirplaneLD-PT-0050

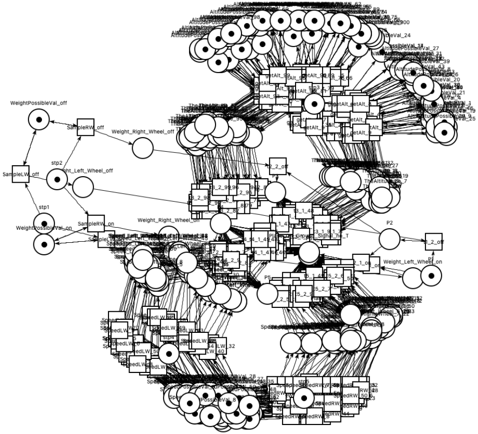


$\equiv_E$

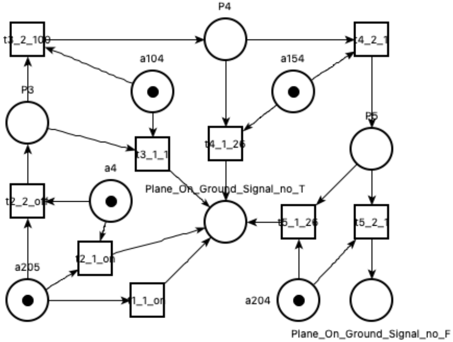


E contains about 400 variables and literals

AirplaneLD-PT-0050

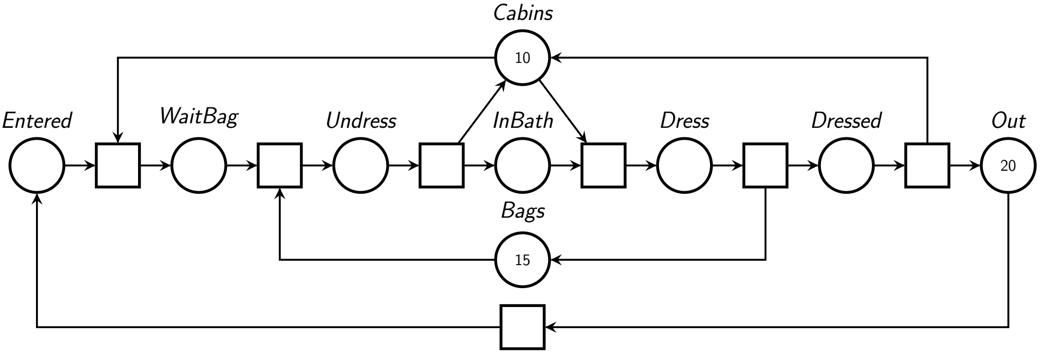


$\equiv_E$



AirplaneLD-PT-**4000**: 30 000 variables and literals

SwimmingPool



$$E \triangleq \begin{cases} \mathit{Cabins} + \mathit{Dress} + \mathit{Dressed} + \mathit{Undress} + \mathit{WaitBag} = 10 \\ \mathit{Dress} + \mathit{Dressed} + \mathit{Entered} + \mathit{InBath} + \mathit{Out} + \mathit{Undress} + \mathit{WaitBag} = 20 \\ \mathit{Bags} + \mathit{Dress} + \mathit{InBath} + \mathit{Undress} = 15 \end{cases}$$

Benchmark (Model Checking Contest)

The Model Checking Contest is important in my work:

- A great source of model instances! $\approx$ 1 400 nets
- Also a source of reachability formulas $\approx$ 50 000 queries

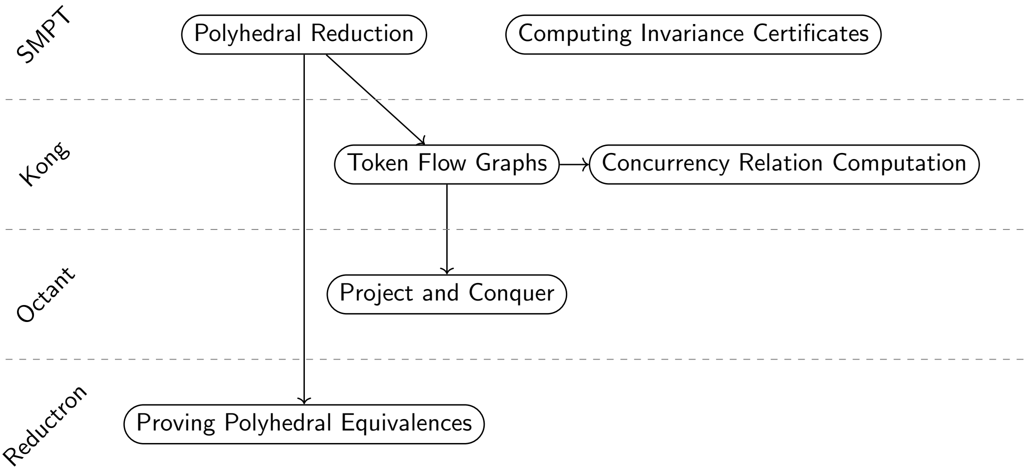
Benchmark (Model Checking Contest)

The Model Checking Contest is important in my work:

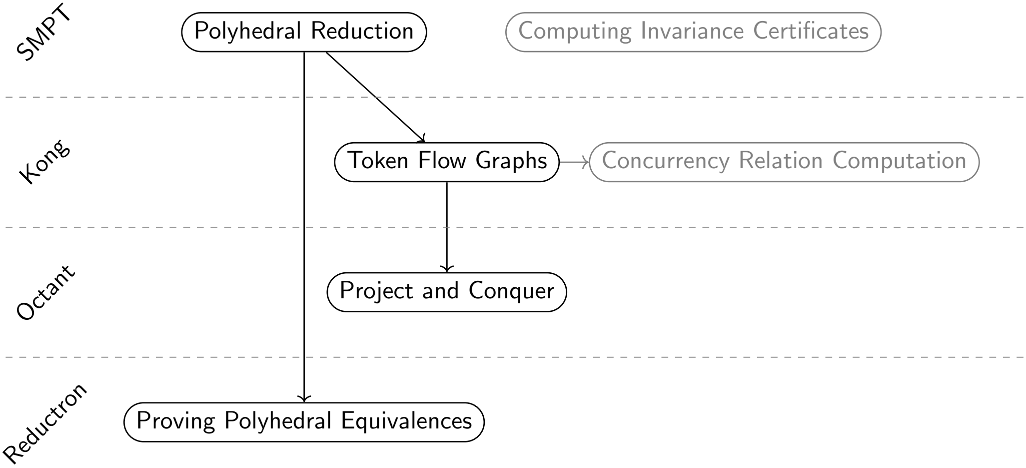
- A great source of model instances! $\approx 1\,400$ nets
- Also a source of reachability formulas $\approx 50\,000$ queries
- **Software development**: from prototypes to tools that can be reused by others

Outline

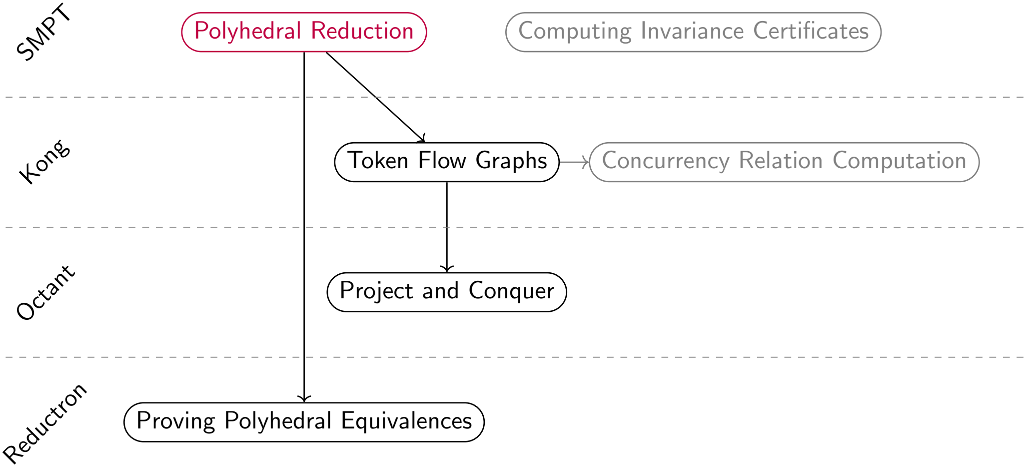
Outline


SMPT
Kong
Octant
Reductron
Polyhedral Reduction
Computing Invariance Certificates
Token Flow Graphs
Concurrency Relation Computation
Project and Conquer
Proving Polyhedral Equivalences

Outline

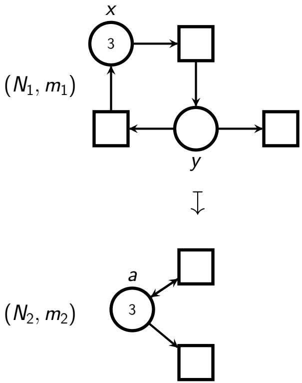

SMPT
Kong
Octant
Reductron
Polyhedral Reduction
Computing Invariance Certificates
Token Flow Graphs
Concurrency Relation Computation
Project and Conquer
Proving Polyhedral Equivalences

Outline

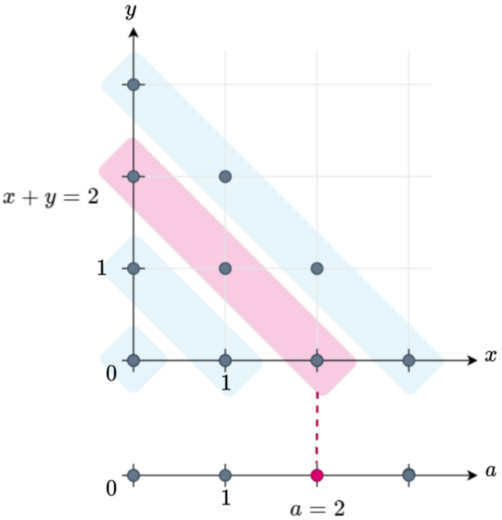

SMPT
Polyhedral Reduction
Computing Invariance Certificates
Kong
Token Flow Graphs
Concurrency Relation Computation
Octant
Project and Conquer
Reductron
Proving Polyhedral Equivalences

Big picture

Polyhedral Reduction



Net reduction example, with $E : a = x + y$



Relation between state-spaces

Markings equivalence up-to E

Polyhedral Reduction

- Two markings m_1 and m_2 are **compatible**:

$$m_1(p) = m_2(p) \text{ for all } p \text{ in } P_1 \cap P_2$$

Markings equivalence up-to E

Polyhedral Reduction

- Two markings m_1 and m_2 are **compatible**:

$$m_1(p) = m_2(p) \text{ for all } p \text{ in } P_1 \cap P_2$$

- A marking m can be associated to **system of equations** $\underline{m}$ defined as:

$$p_1 = m(p_1) \wedge \cdots \wedge p_k = m(p_k) \text{ where } P = \{p_1, \ldots, p_k\}$$

Markings equivalence up-to E

Polyhedral Reduction

- Two markings m_1 and m_2 are **compatible**:

$$m_1(p) = m_2(p) \text{ for all } p \text{ in } P_1 \cap P_2$$

- A marking m can be associated to **system of equations** $\underline{m}$ defined as:

$$p_1 = m(p_1) \wedge \cdots \wedge p_k = m(p_k) \text{ where } P = \{p_1, \ldots, p_k\}$$

- We denote $m_1 \equiv_E m_2$ when:

$$E \wedge \underline{m_1} \wedge \underline{m_2} \text{ is satisfiable}$$

Polyhedral equivalence

Polyhedral Reduction

Definition (Relaxed E-equivalence)

$(N_1, m_1) \equiv_E (N_2, m_2)$ if and only if

(A1) initial markings are *realated up-to* E: $m_1 \equiv_E m_2$;

(A2a) for all markings m in $R(N_1, m_1)$ or $R(N_2, m_2)$: $E \wedge \underline{m}$ is satisfiable;

(A2b) assume m'_1, m'_2 are markings of N_1, N_2 related up-to E, such that $m'_1 \equiv_E m'_2$, then m'_1 is reachable iff m'_2 is reachable.

Polyhedral equivalence

Polyhedral Reduction

Definition (Relaxed E-equivalence)

$(N_1, m_1) \equiv_E (N_2, m_2)$ if and only if

- **(A1)** initial markings are *realated up-to* E: $m_1 \equiv_E m_2$;
- **(A2a)** for all markings m in $R(N_1, m_1)$ or $R(N_2, m_2)$: $E \wedge \underline{m}$ is satisfiable;
- **(A2b)** assume m'_1, m'_2 are markings of N_1, N_2 related up-to E, such that $m'_1 \equiv_E m'_2$, then m'_1 is reachable iff m'_2 is reachable.

We have two variant definitions:

- Composition (relies on observation sequences)
- Automated proving

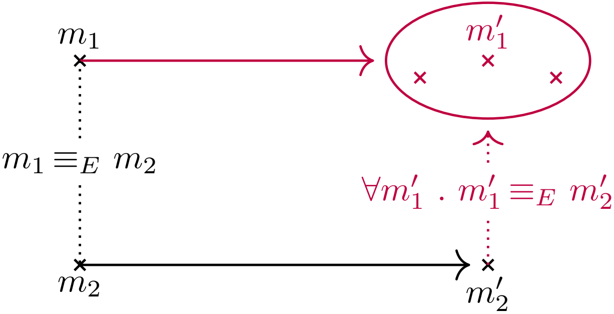
Key results: reachability checking

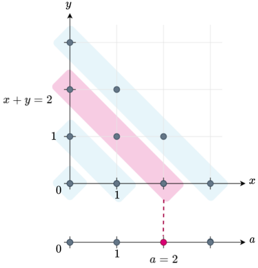
Polyhedral Reduction

Lemma (Reachability checking)

For all pairs of markings m'_1, m'_2 of N_1, N_2 such that $m'_1 \equiv_E m'_2$:

$$\textit{if } m'_2 \in R(N_2, m_2) \textit{ then } m'_1 \in R(N_1, m_1).$$



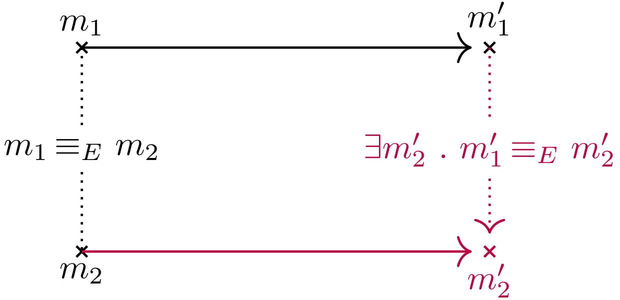


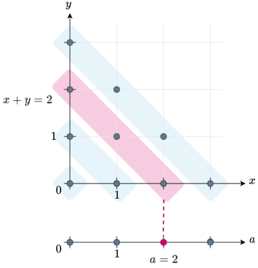
Key results: invariance checking

Polyhedral Reduction

Lemma (Invariance checking)

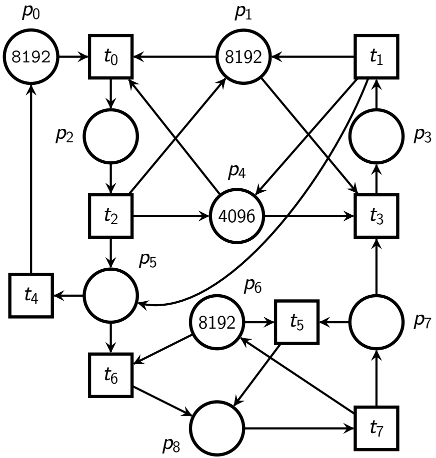
For all m'_1 in $R(N_1, m_1)$ there is m'_2 in $R(N_2, m_2)$ such that $m'_1 \equiv_E m'_2$.





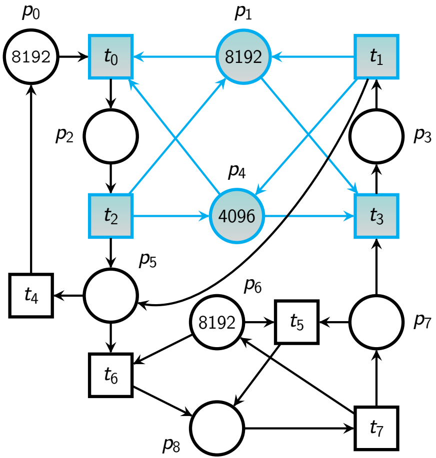
Deriving polyhedral reductions – Step 1

Polyhedral Reduction



Deriving polyhedral reductions – Step 1

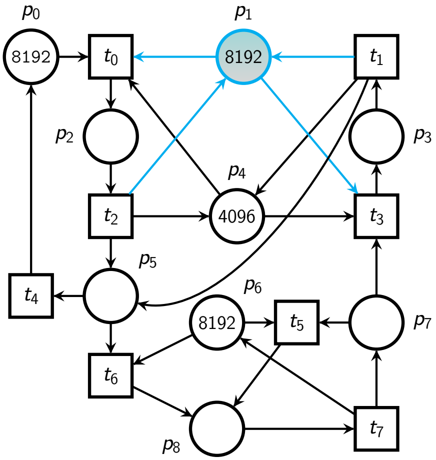
Polyhedral Reduction



Rule [RED]: place p_1 is redundant to p_4

Deriving polyhedral reductions – Step 1

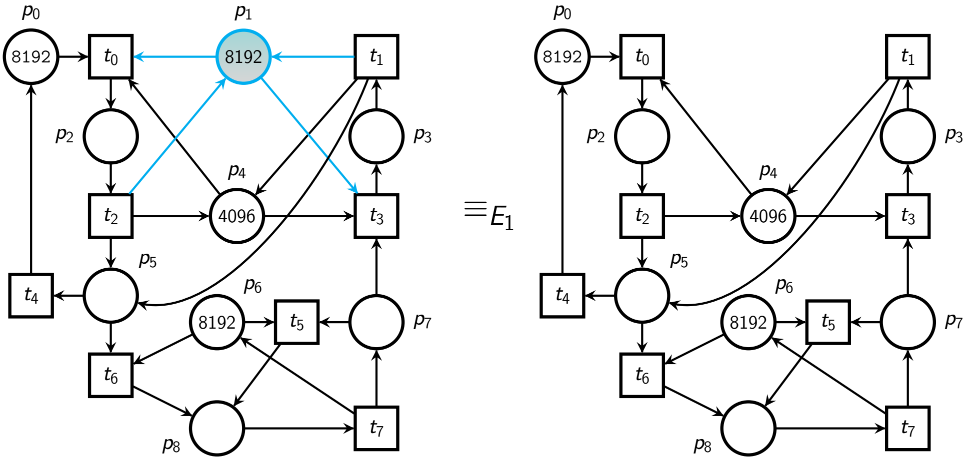
Polyhedral Reduction



Rule [RED]: place p_1 is redundant to p_4

Deriving polyhedral reductions – Step 1

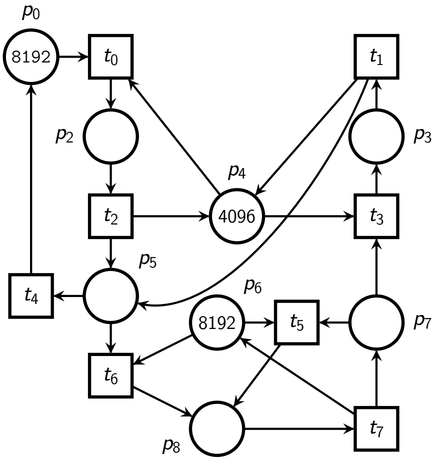
Polyhedral Reduction



$$E_1 \triangleq p_1 = p_4 + 4096$$

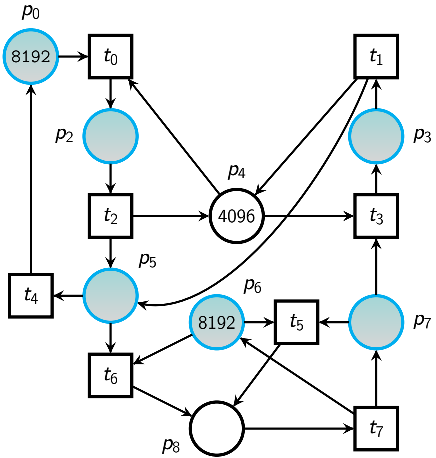
Deriving polyhedral reductions – Step 2

Polyhedral Reduction



Deriving polyhedral reductions – Step 2

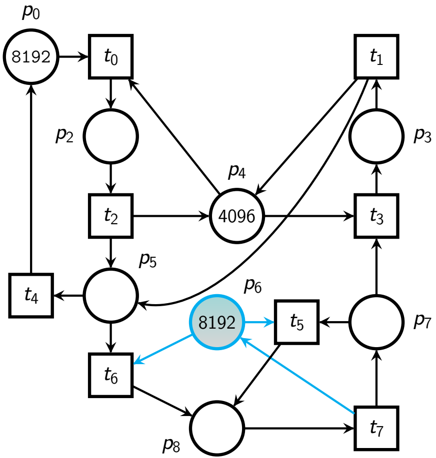
Polyhedral Reduction



Place invariant: $p_6 = p_0 + p_2 + p_3 + p_5 + p_7$

Deriving polyhedral reductions – Step 2

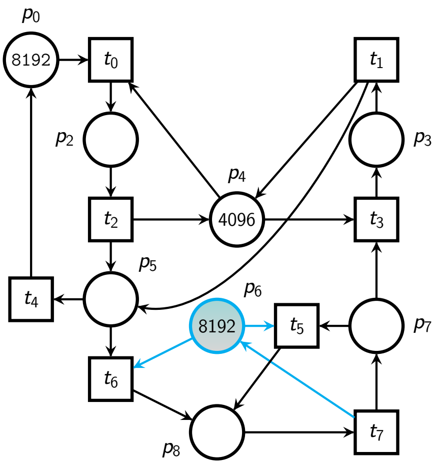
Polyhedral Reduction



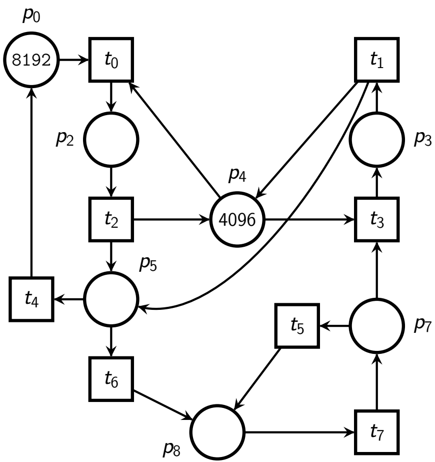
Place invariant: $p_6 = p_0 + p_2 + p_3 + p_5 + p_7$

Deriving polyhedral reductions – Step 2

Polyhedral Reduction



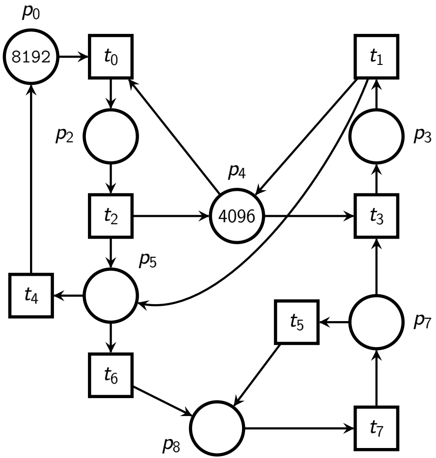
$\equiv_{E_2}$



$$E_2 \triangleq p_6 = p_0 + p_2 + p_3 + p_5 + p_7$$

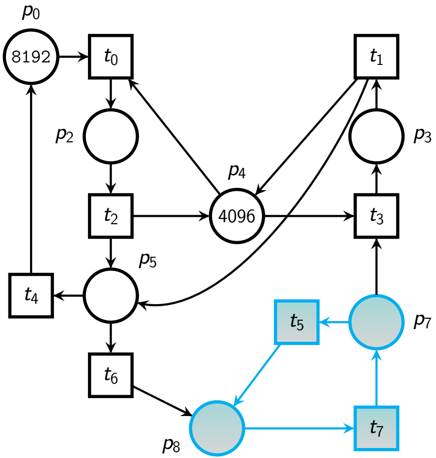
Deriving polyhedral reductions – Step 3

Polyhedral Reduction



Deriving polyhedral reductions – Step 3

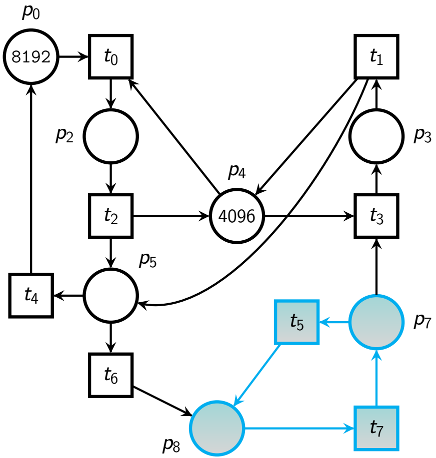
Polyhedral Reduction



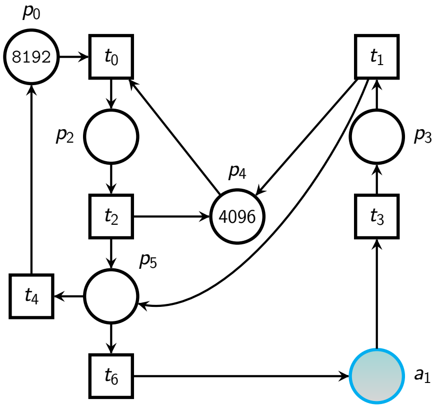
Rule [AGG]: agglomerate places p_7 and p_8 into a new place

Deriving polyhedral reductions – Step 3

Polyhedral Reduction



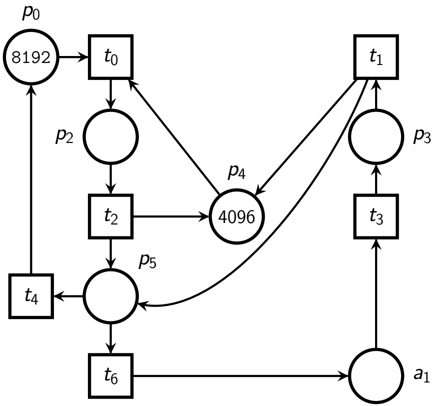
$\equiv_{E_3}$



$$E_3 \triangleq a_1 = p_7 + p_8$$

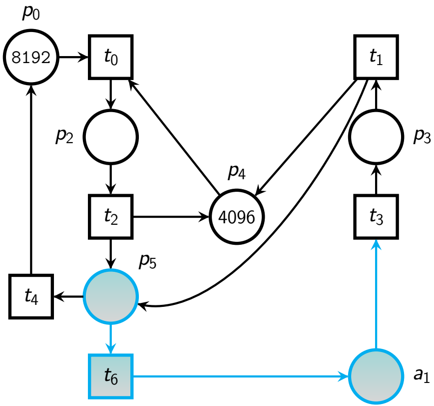
Deriving polyhedral reductions – Step 4

Polyhedral Reduction



Deriving polyhedral reductions – Step 4

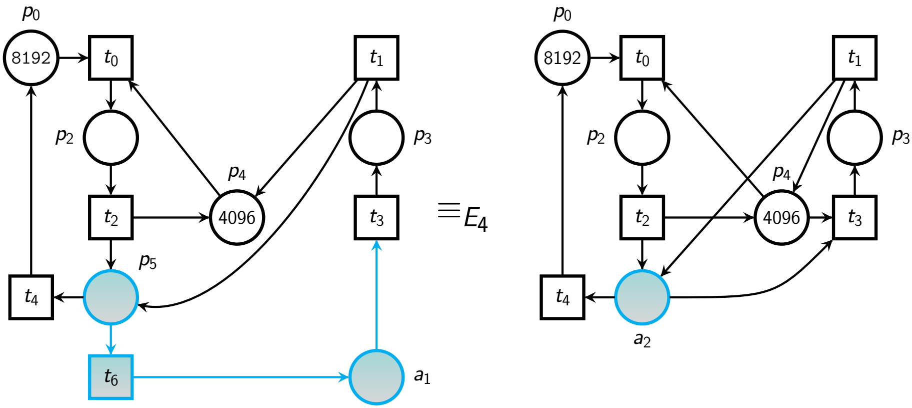
Polyhedral Reduction



Rule [CONCAT]: concatenate a_1 and p_5 into a new place

Deriving polyhedral reductions – Step 4

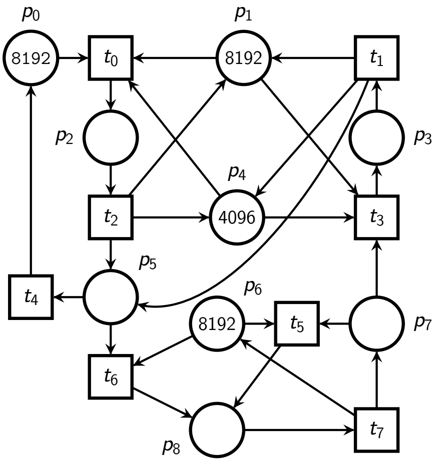
Polyhedral Reduction



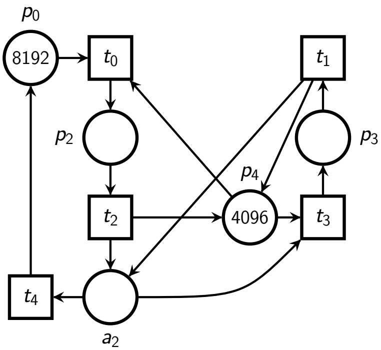
$$E_4 \triangleq a_2 = a_1 + p_5$$

Deriving polyhedral reductions – Step 4

Polyhedral Reduction



$\equiv_E$



$$E \triangleq \exists a_1. \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_6 \end{cases}$$

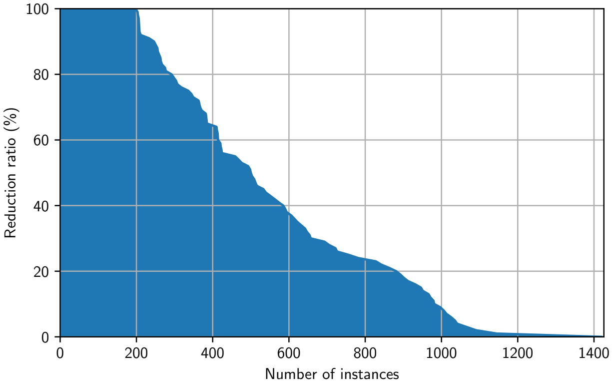
Reduction rules: $[\text{RED}]$, $[\text{AGG}]$, $[\text{CONCAT}]$, ...

Laws:

- Composability (congruence for $\|$-composition)
- Transitivity
- Relabeling

Prevalence of reductions over the 1 426 MCC instances

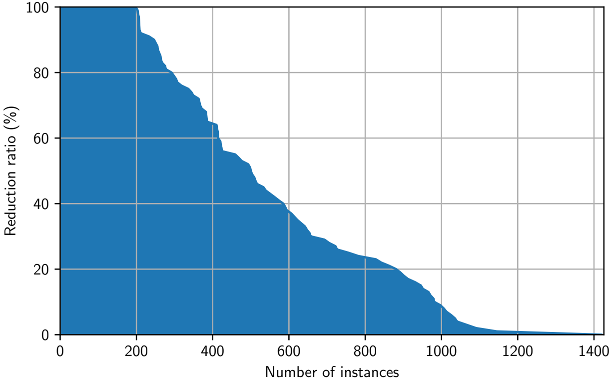
Polyhedral Reduction



- 80% of instances are reduced by $> 1\%$
- Half of them are significantly reduced (reduction ratio $> 30\%$)
- 14% of fully reducible instances

Prevalence of reductions over the 1 426 MCC instances

Polyhedral Reduction



How to combine with the reachability problem?

Combination with reachability

Polyhedral Reduction

- Is F_1 reachable in (N_1, m_1)?

$$F_1 \triangleq \begin{cases} 3p_7 + 2p_8 & \geqslant p_6 \\ p_8 & \geqslant p_1 \end{cases}$$

Combination with reachability

Polyhedral Reduction

- Is F_1 reachable in (N_1, m_1)?

$$F_1 \triangleq \begin{cases} 3p_7 + 2p_8 & \geqslant p_6 \\ p_8 & \geqslant p_1 \end{cases}$$

Definition (E-Transform Formula)

Formula $F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{p_1}.\, \tilde{E}(\boldsymbol{p_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{p_1})$ is the E-transform of F_1.

Combination with reachability

Polyhedral Reduction

- Is F_1 reachable in (N_1, m_1)?

$$F_1 \triangleq \begin{cases} 3p_7 + 2p_8 & \geqslant p_6 \\ p_8 & \geqslant p_1 \end{cases}$$

Definition (E-Transform Formula)

Formula $F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{q_1}.\, \tilde{E}(\boldsymbol{q_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{q_1})$ is the E-transform of F_1.

$$F_2 \triangleq \exists q_0, .., q_8.\, \exists a_1.\, \begin{cases} q_1 = q_4 + 4096 \\ q_6 = q_0 + q_2 + q_3 + q_5 + q_7 \\ a_1 = q_7 + q_8 \\ a_2 = a_1 + q_6 \end{cases} \wedge \begin{cases} p_0 = q_0 \\ p_2 = q_2 \\ p_3 = q_3 \\ p_4 = q_4 \end{cases} \wedge \begin{cases} 3q_7 + 2q_8 & \geqslant q_6 \\ q_8 & \geqslant q_1 \end{cases}$$

Combination with reachability

Polyhedral Reduction

- Is F_1 reachable in (N_1, m_1)? $\quad F_1 \triangleq \begin{cases} 3p_7 + 2p_8 & \geqslant p_6 \\ p_8 & \geqslant p_1 \end{cases}$

> **Definition (E-Transform Formula)**
> Formula $F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{q_1}.\, \tilde{E}(\boldsymbol{q_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{q_1})$ is the E-transform of F_1.

$$F_2 \triangleq \exists q_0, .., q_8.\, \exists a_1.\, \begin{cases} q_1 = q_4 + 4096 \\ q_6 = q_0 + q_2 + q_3 + q_5 + q_7 \\ a_1 = q_7 + q_8 \\ a_2 = a_1 + q_6 \end{cases} \wedge \begin{cases} p_0 = q_0 \\ p_2 = q_2 \\ p_3 = q_3 \\ p_4 = q_4 \end{cases} \wedge \begin{cases} 3q_7 + 2q_8 & \geqslant q_6 \\ q_8 & \geqslant q_1 \end{cases}$$

- Is the E-transform formula F_2 reachable in (N_2, m_2)?

Fundamental results on E-transform formulas

Polyhedral Reduction

Theorem (Reachability Conservation)

F_1 is reachable in N_1 if and only if its E-transform formula F_2 is reachable in N_2.

Fundamental results on E-transform formulas

Polyhedral Reduction

Theorem (Reachability Conservation)

F_1 is reachable in N_1 if and only if its E-transform formula F_2 is reachable in N_2.

Corollary (Invariant Conservation)

$\neg F_1$ invariant on N_1 if and only if $\neg F_2$ invariant on N_2.

Fundamental results on E-transform formulas

Polyhedral Reduction

Theorem (Reachability Conservation)

F_1 is reachable in N_1 if and only if its E-transform formula F_2 is reachable in N_2.

Corollary (Invariant Conservation)

$\neg F_1$ invariant on N_1 if and only if $\neg F_2$ invariant on N_2.

Does it fit well with SMT-based methods?

Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p^{(0)}})$

Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{\mathbf{(0)}})$ $\qquad\qquad$ $\phi_0 \land F(\boldsymbol{p}^{\mathbf{(0)}})$ `sat?`

Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(\mathbf{0})})$

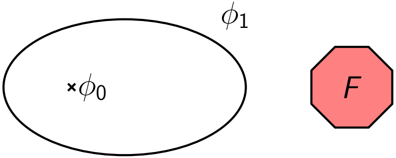
$\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(\mathbf{0})})}$ ~~sat~~ unsat

Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(0)})$
2. $\phi_1 \triangleq \phi_0 \wedge \mathrm{T}(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)})$

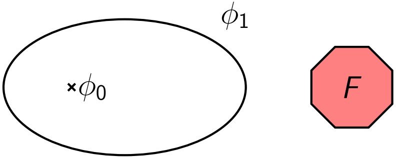
$\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(0)})\ \texttt{sat}}$ `unsat`



Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(0)})$ — $\cancel{\phi_0 \land F(\boldsymbol{p}^{(0)}) \texttt{ sat}}$ $\texttt{unsat}$
2. $\phi_1 \triangleq \phi_0 \land \mathrm{T}(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)})$ — $\phi_1 \land F(\boldsymbol{p}^{(1)})$ $\texttt{sat?}$

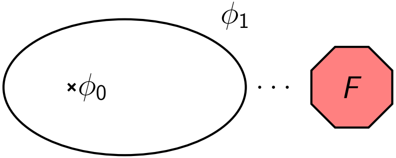


Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p^{(0)}})$ $\cancel{\phi_0 \wedge F(\boldsymbol{p^{(0)}}) \texttt{ sat}}$ $\texttt{unsat}$
2. $\phi_1 \triangleq \phi_0 \wedge \mathrm{T}(\boldsymbol{p^{(0)}}, \boldsymbol{p^{(1)}})$ $\cancel{\phi_0 \wedge F(\boldsymbol{p^{(1)}}) \texttt{ sat}}$ $\texttt{unsat}$

   $\ldots$



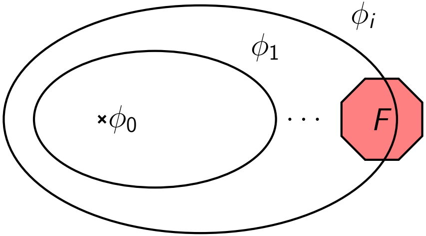
Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(\mathbf{0})})$ — $\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(\mathbf{0})})\ \texttt{sat}}$ $\texttt{unsat}$
2. $\phi_1 \triangleq \phi_0 \wedge \mathrm{T}(\boldsymbol{p}^{(\mathbf{0})}, \boldsymbol{p}^{(\mathbf{1})})$ — $\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(\mathbf{1})})\ \texttt{sat}}$ $\texttt{unsat}$

   $\ldots$

3. $\phi_i \triangleq \phi_{i-1} \wedge \mathrm{T}(\boldsymbol{p}^{(\boldsymbol{i}-\mathbf{1})}, \boldsymbol{p}^{(\boldsymbol{i})})$



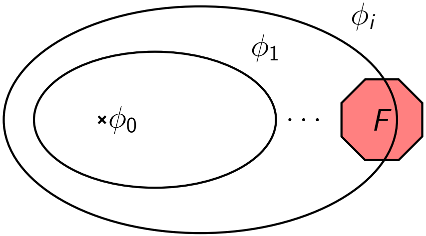
Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(0)})$ $\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(0)}) \texttt{ sat}}$ $\texttt{unsat}$
2. $\phi_1 \triangleq \phi_0 \wedge \mathrm{T}(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)})$ $\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(1)}) \texttt{ sat}}$ $\texttt{unsat}$

   $\ldots$

3. $\phi_i \triangleq \phi_{i-1} \wedge \mathrm{T}(\boldsymbol{p}^{(i-1)}, \boldsymbol{p}^{(i)})$ $\phi_i \wedge F(\boldsymbol{p}^{(i)})$ $\texttt{sat}$



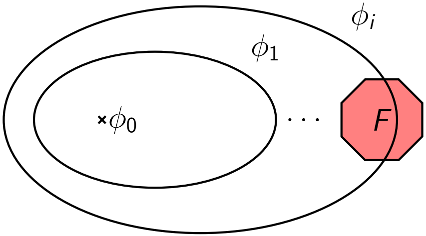
Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(0)})$ $\qquad$ $\cancel{\phi_0 \land F(\boldsymbol{p}^{(0)})\ \texttt{sat}}$ $\texttt{unsat}$
2. $\phi_1 \triangleq \phi_0 \land \mathrm{T}(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)})$ $\qquad$ $\cancel{\phi_0 \land F(\boldsymbol{p}^{(1)})\ \texttt{sat}}$ $\texttt{unsat}$

   ...

3. $\phi_i \triangleq \phi_{i-1} \land \mathrm{T}(\boldsymbol{p}^{(i-1)}, \boldsymbol{p}^{(i)})$ $\qquad$ $\phi_i \land F(\boldsymbol{p}^{(i)})$ $\texttt{sat}$



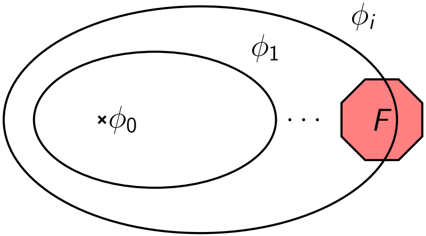
If $\phi_i(N_1) \land F_1$ sat in N_1 then there is $j \leqslant i$ such that $\phi_j(N_2) \land F_2$ sat in N_2

Bounded Model Checking (BMC) [Biere, 99]

Polyhedral Reduction

1. $\phi_0 \triangleq \underline{m_0}(\boldsymbol{p}^{(0)})$ — $\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(0)})\ \texttt{sat}}\ \texttt{unsat}$
2. $\phi_1 \triangleq \phi_0 \wedge \mathrm{T}(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)})$ — $\cancel{\phi_0 \wedge F(\boldsymbol{p}^{(1)})\ \texttt{sat}}\ \texttt{unsat}$

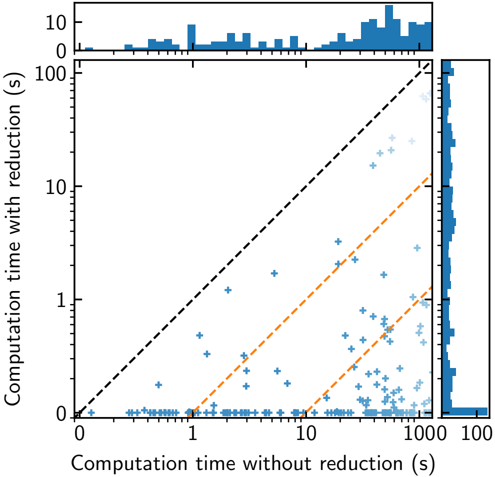
   $\ldots$
3. $\phi_i \triangleq \phi_{i-1} \wedge \mathrm{T}(\boldsymbol{p}^{(i-1)}, \boldsymbol{p}^{(i)})$ — $\phi_i \wedge F(\boldsymbol{p}^{(i)})\ \texttt{sat}$



If $\phi_i(N_1) \wedge F_1$ sat in N_1 then there is $j \ll i$ such that $\phi_j(N_2) \wedge F_2$ sat in N_2

Performance evaluation: 50% $\leqslant$ reduction ratio $<$ 100%

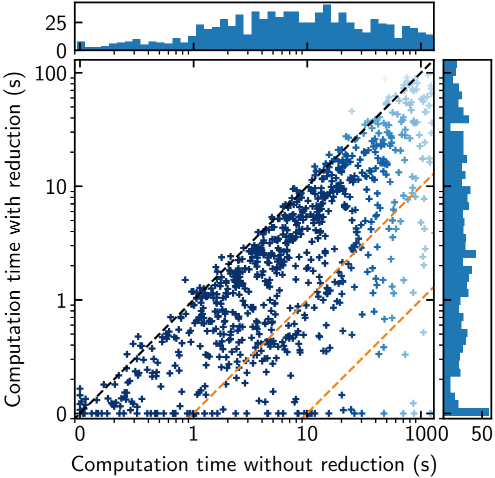
Polyhedral Reduction



$\times$2.6 computed queries

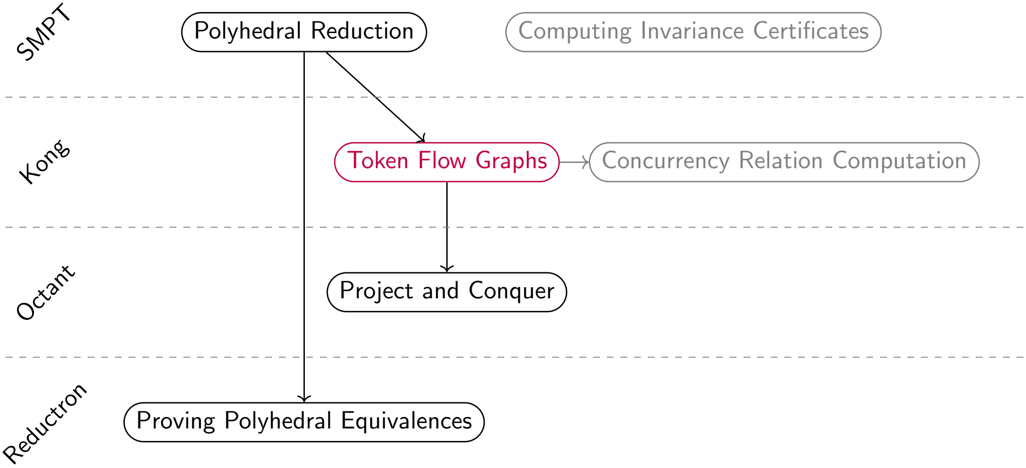
Performance evaluation: $1\% \leqslant$ reduction ratio $< 25\%$

Polyhedral Reduction



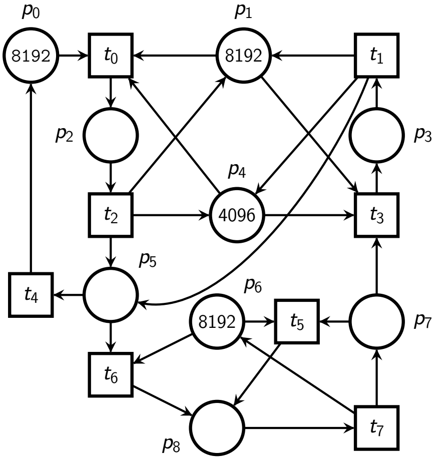
$\times$1.22 computed queries

Outline

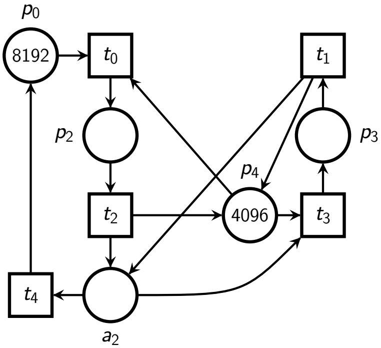

SMPT
Polyhedral Reduction
Computing Invariance Certificates
Kong
Token Flow Graphs
Concurrency Relation Computation
Octant
Project and Conquer
Reductron
Proving Polyhedral Equivalences

SmallOperatingSystem

Token Flow Graphs



$\equiv_E$



$$E \triangleq \exists a_1. \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$

Motivation

Token Flow Graphs

- Reason on **graphs** instead of solving **Presburger formulas**

- Capture the **particular structure** of constraints from polyhedral reductions

$$E \triangleq \exists a_1.\ \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$

Motivation

Token Flow Graphs

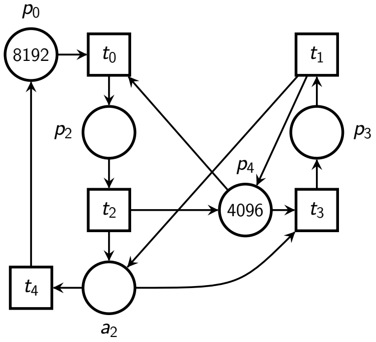
- Reason on **graphs** instead of solving **Presburger formulas**
- Capture the **particular structure** of constraints from polyhedral reductions
- Directed Acyclic Graph (**DAG**) with two kinds of arcs

$$E \triangleq \exists a_1. \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$

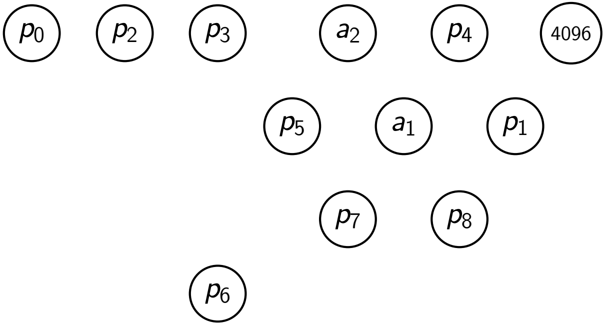
Construction

Token Flow Graphs

$$\exists a_1. \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$



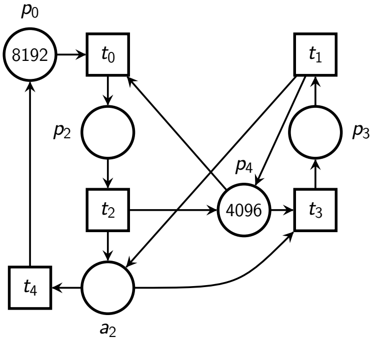
(N_2, m_2)



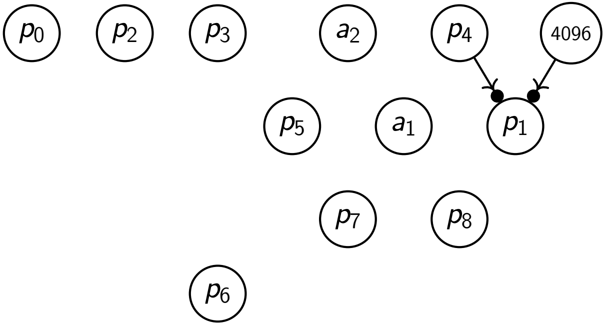
Construction

Token Flow Graphs

$$\exists a_1 . \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$



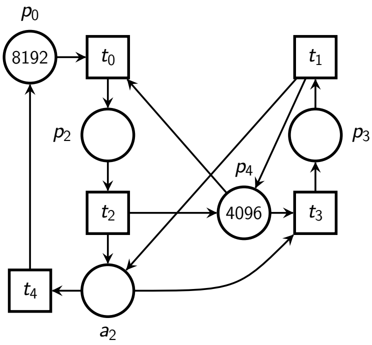
(N_2, m_2)



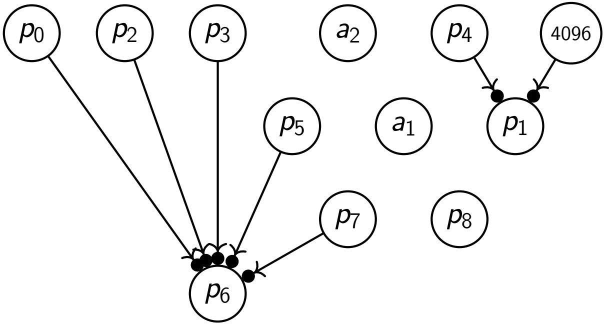
Construction

Token Flow Graphs

$$\exists a_1 . \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$



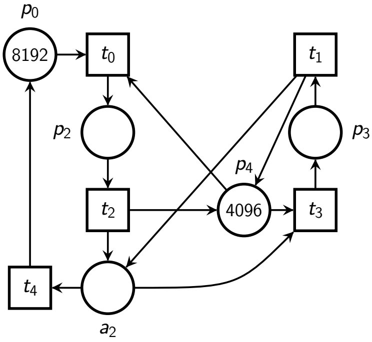
(N_2, m_2)



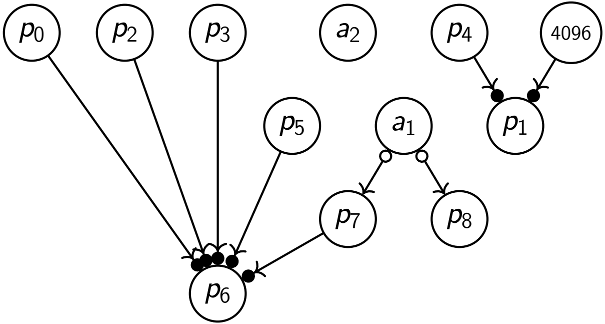
Construction

Token Flow Graphs

$$\exists a_1 . \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$



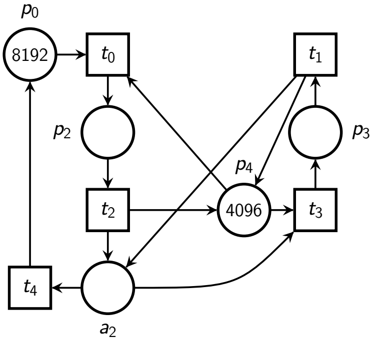
(N_2, m_2)



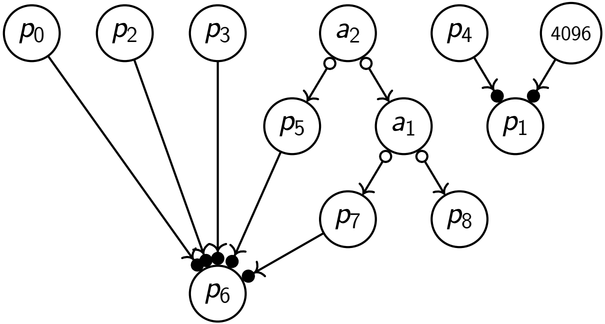
Construction

Token Flow Graphs

$$\exists a_1.\begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$



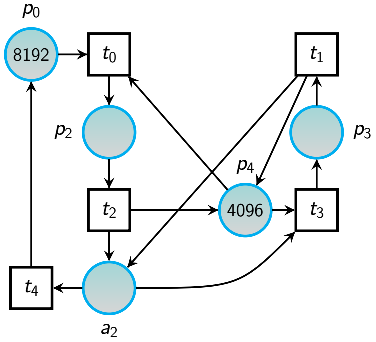
(N_2, m_2)



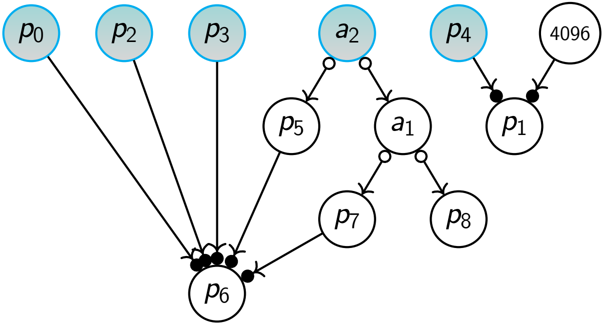
Construction

Token Flow Graphs

$$\exists a_1 . \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$

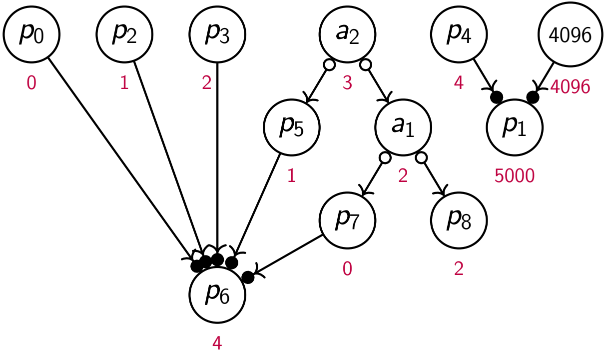


(N_2, m_2)



Configuration of a TFG

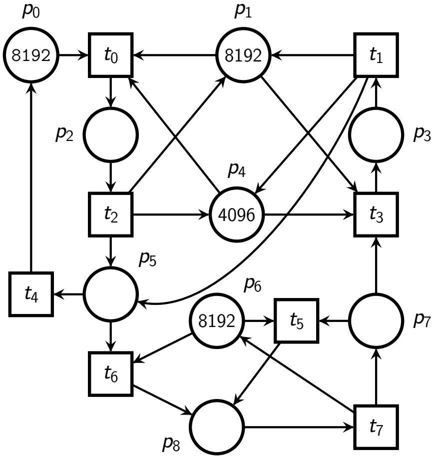
Token Flow Graphs



- **Configuration** c: partial function from set of nodes V to $\mathbb{N}$
- **Well-defined**: $\underline{c} \land E$ is satisfiable
- **Total**: defined for all nodes

Configuration reachability

Token Flow Graphs

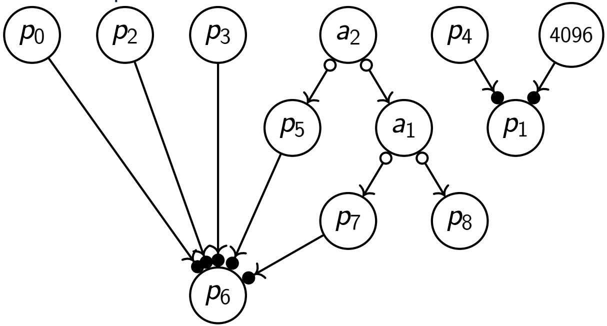


$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

Is m' reachable from the initial marking?

Configuration reachability

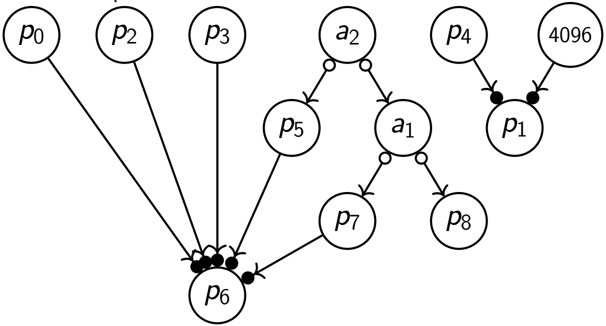
Token Flow Graphs



$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

Configuration reachability

Token Flow Graphs



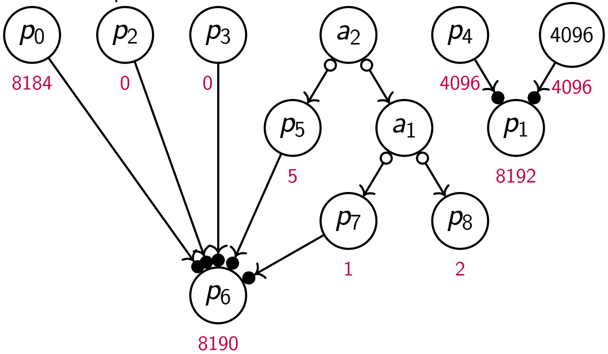
$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

Theorem (Reachable marking extension and unicity)

If m' is a marking in $R(N_1, m_1)$ then there exists a unique, total and well-defined configuration c of $[\![E]\!]$ such that $c_{|N_1} = m$.

Configuration reachability

Token Flow Graphs



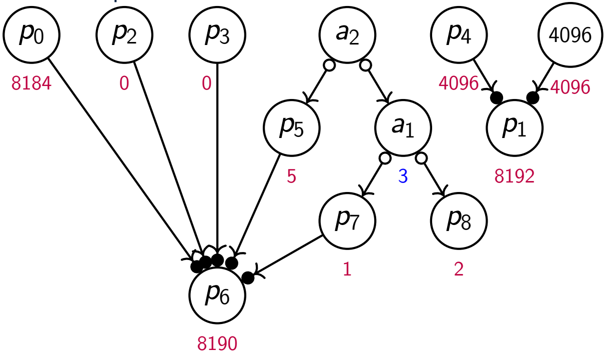
$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

Theorem (Reachable marking extension and unicity)

If m' is a marking in $R(N_1, m_1)$ then there exists a unique, total and well-defined configuration c of $[\![E]\!]$ such that $c_{|N_1} = m$.

Configuration reachability

Token Flow Graphs



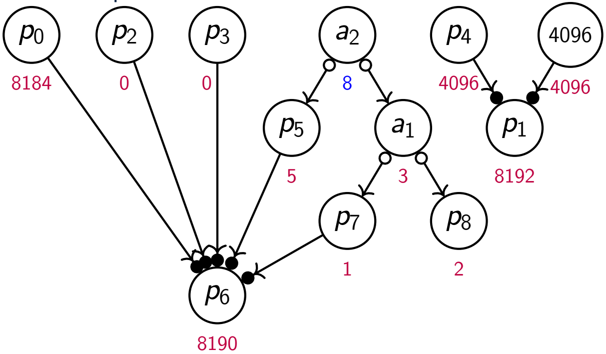
$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

Theorem (Reachable marking extension and unicity)

If m' is a marking in $R(N_1, m_1)$ then there exists a unique, total and well-defined configuration c of $[\![E]\!]$ such that $c_{|N_1} = m$.

Configuration reachability

Token Flow Graphs



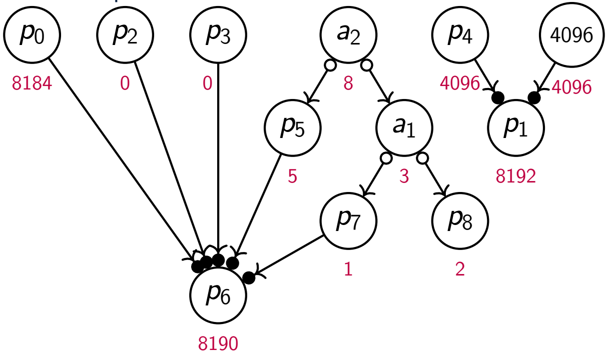
$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

Theorem (Reachable marking extension and unicity)

If m' is a marking in $R(N_1, m_1)$ then there exists a unique, total and well-defined configuration c of $[\![E]\!]$ such that $c_{|N_1} = m$.

Configuration reachability

Token Flow Graphs



$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

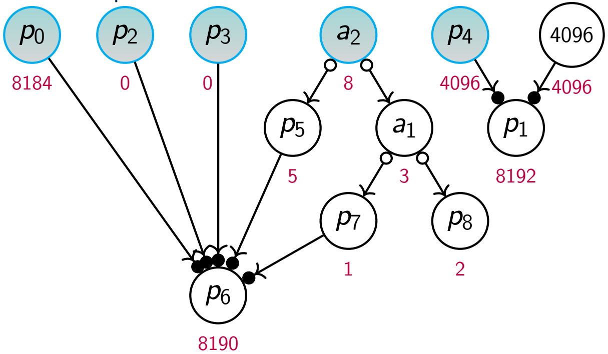
Theorem (Reachable marking extension and unicity)

If m' is a marking in $R(N_1, m_1)$ then there exists a unique, total and well-defined configuration c of $[\![E]\!]$ such that $c_{|N_1} = m$.

Corollary: if c does not exist then m' not reachable

Configuration reachability

Token Flow Graphs



$$m' \triangleq \begin{cases} p_0 = 8184 \\ p_1 = 8192 \\ p_2 = 0 \\ p_3 = 0 \\ p_4 = 4096 \\ p_5 = 5 \\ p_6 = 8190 \\ p_7 = 1 \\ p_8 = 2 \end{cases}$$

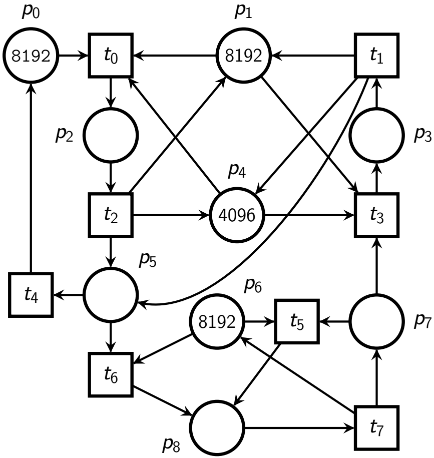
Theorem (Reachability equivalence)

Given a total, well-defined configuration c:

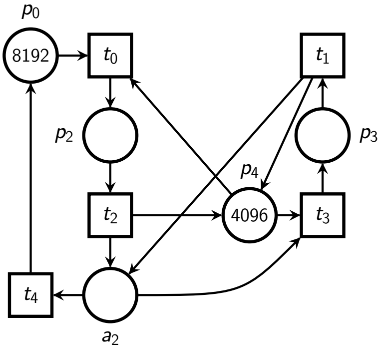
$$c_{|N_2} \in R(N_2, m_2) \textit{ if and only if } c_{|N_1} \in R(N_1, m_1)$$

Non-TFGizable polyhedral reduction

Token Flow Graphs



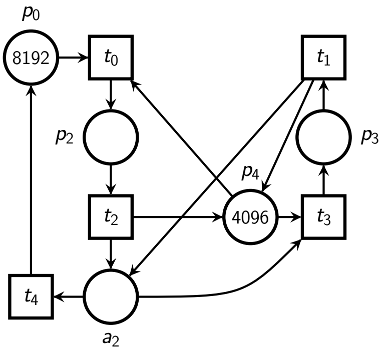
$\equiv_E$



$$E \triangleq \exists a_1.\ \begin{cases} p_1 = p_4 + 4096 \\ p_6 = p_0 + p_2 + p_3 + p_5 + p_7 \\ a_1 = p_7 + p_8 \\ a_2 = a_1 + p_5 \end{cases}$$

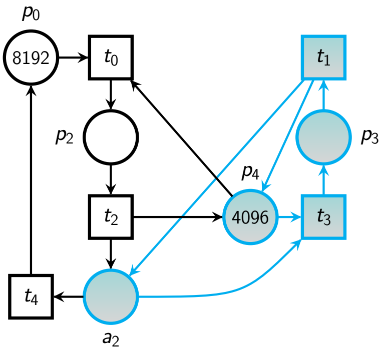
Non-TFGizable polyhedral reduction

Token Flow Graphs



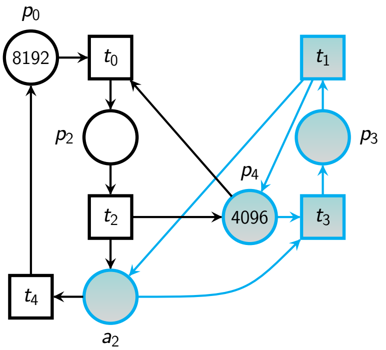
Non-TFGizable polyhedral reduction

Token Flow Graphs

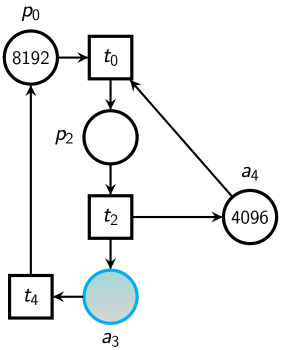


Non-TFGizable polyhedral reduction

Token Flow Graphs



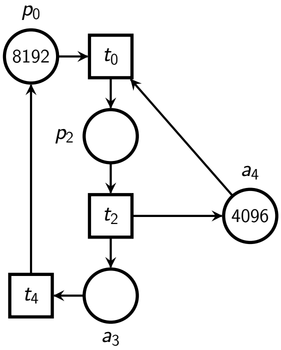
$\equiv_{E_5}$



$$E_5 \triangleq \begin{cases} a_3 = a_2 + p_3 \\ a_4 = p_4 + p_3 \end{cases}$$

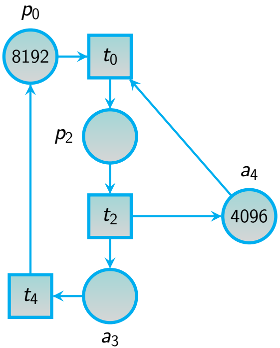
Non-TFGizable polyhedral reduction

Token Flow Graphs



Non-TFGizable polyhedral reduction

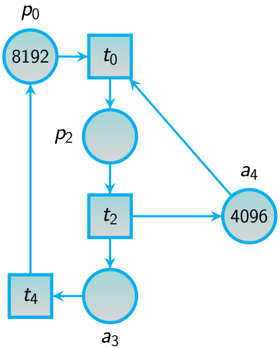
Token Flow Graphs



Live Marked Graph: state equation is exact!

Non-TFGizable polyhedral reduction

Token Flow Graphs

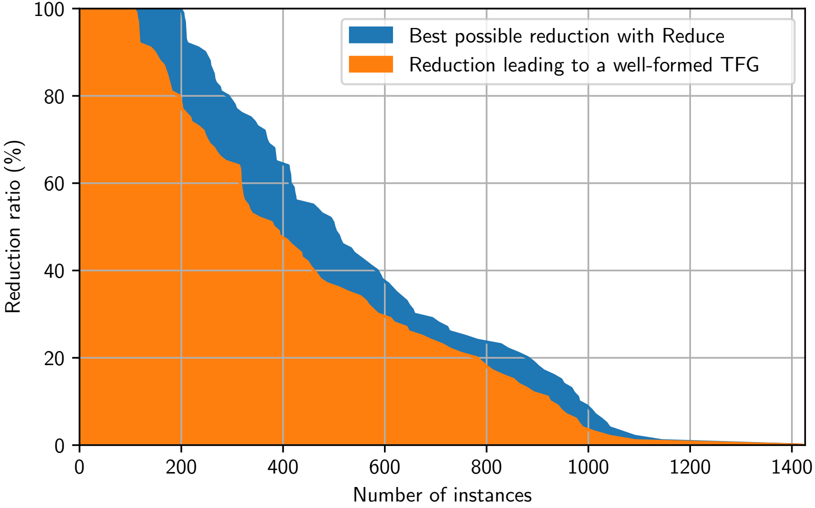


$$\equiv_{E_6} \quad \emptyset$$

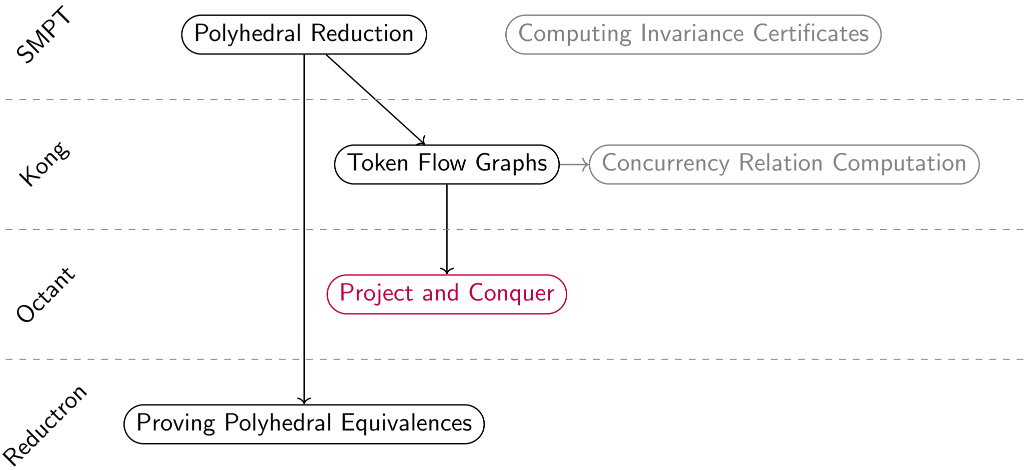
$$E_6 \triangleq \begin{cases} a_3 + p_0 + p_2 & = 8192 \\ p_2 + a_4 & = 4096 \end{cases}$$

Prevalence of reductions over the MCC instances

Token Flow Graphs



Outline


SMPT
Polyhedral Reduction
Computing Invariance Certificates
Kong
Token Flow Graphs
Concurrency Relation Computation
Octant
Project and Conquer
Reductron
Proving Polyhedral Equivalences

Previous context

Project and Conquer

Definition (E-Transform Formula)

$F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{p_1}.\, \tilde{E}(\boldsymbol{p_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{p_1})$ is the E-transform of F_1

Definition (E-Transform Formula)

$F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{p_1}.\, \tilde{E}(\boldsymbol{p_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{p_1})$ is the E-transform of F_1

Theorem (Reachability Conservation)

F_1 reachable in N_1 if and only if F_2 reachable in N_2

Definition (E-Transform Formula)

$F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{p_1}.\, \tilde{E}(\boldsymbol{p_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{p_1})$ is the E-transform of F_1

Theorem (Reachability Conservation)

F_1 reachable in N_1 if and only if F_2 reachable in N_2

- **Not suitable with random exploration**
  (need to evaluate a quantified formula for each visited state)
- **Not usable with standard model-checkers**
  (only support quantifier-free formulas on the set of places)

Previous context

Project and Conquer

Definition (E-Transform Formula)

$F_2(\boldsymbol{p_2}) \triangleq \exists \boldsymbol{p_1} .\, \tilde{E}(\boldsymbol{p_1}, \boldsymbol{p_2}) \wedge F_1(\boldsymbol{p_1})$ is the E-transform of F_1

Theorem (Reachability Conservation)

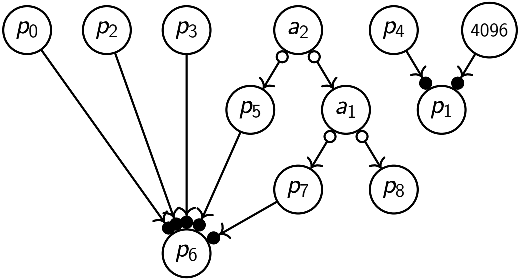
F_1 reachable in N_1 if and only if F_2 reachable in N_2

- **Not suitable with random exploration**
  (need to evaluate a quantified formula for each visited state)
- **Not usable with standard model-checkers**
  (only support quantifier-free formulas on the set of places)

We introduce a procedure to eliminate quantifiers in F_2 (EXPSPACE in general)

Running example

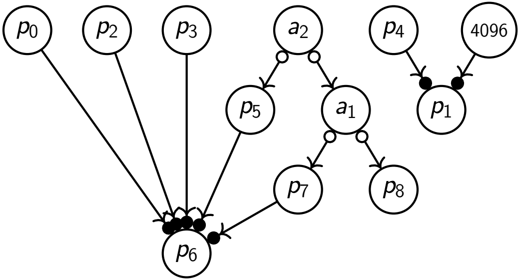
Project and Conquer



$$F_1 \triangleq (3p_7 + 2p_8 \geqslant p_6) \wedge (p_8 \geqslant p_1)$$

Running example

Project and Conquer

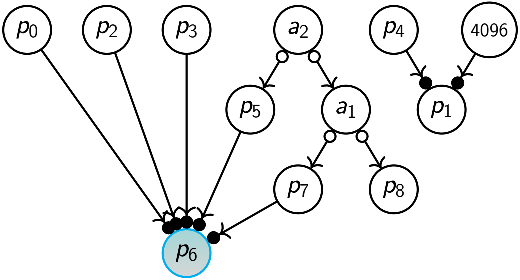


$$
\begin{array}{rcrcrcl}
3\,p_7 & + & 2\,p_8 & - & p_6 & \geqslant & 0 \\
 & & p_8 & - & p_1 & \geqslant & 0
\end{array}
$$

$$\Updownarrow$$

Running example

Project and Conquer

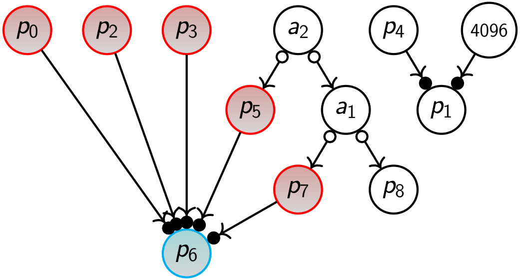


$$\begin{array}{rcrcrl} 3\,p_7 & + & 2\,p_8 & - & \color{blue}{p_6} & \geqslant 0 \\ & & p_8 & - & p_1 & \geqslant 0 \end{array}$$

$$\Updownarrow$$

Running example

Project and Conquer



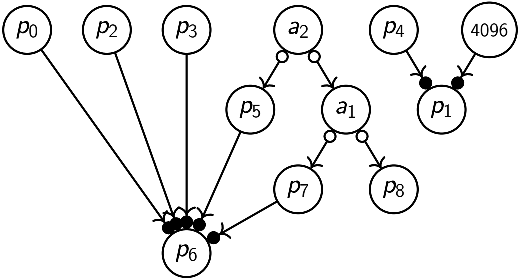
$$\begin{array}{rcrcrl} 3\,p_7 & + & 2\,p_8 & - & p_6 & \geqslant 0 \\ & & p_8 & - & p_1 & \geqslant 0 \end{array}$$

$$\Updownarrow$$

$$\begin{array}{rcrcrl} 3\,p_7 & + & 2\,p_8 & - & (p_0 + p_2 + p_3 + p_5 + p_7) & \geqslant 0 \\ & & p_8 & - & p_1 & \geqslant 0 \end{array}$$

Running example

Project and Conquer



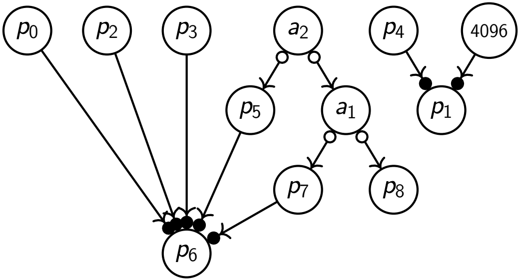
$$\begin{array}{rcrcrl} 3\,p_7 & + & 2\,p_8 & - & p_6 & \geqslant 0 \\ p_8 & - & p_1 & & & \geqslant 0 \end{array}$$

$$\Updownarrow$$

$$\begin{array}{rcrcrcrcrcrl} 2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant 0 \\ & & & & & & & & p_8 & - & p_1 & \geqslant 0 \end{array}$$

Running example

Project and Conquer

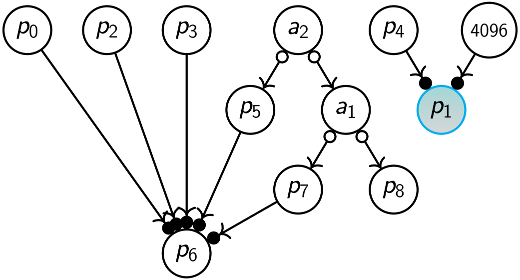


$$\begin{array}{ccccccccccccccc} 2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant 0 \\ & & & & & & & & p_8 & - & p_1 & \geqslant 0 \end{array}$$

$$\Updownarrow$$

Running example

Project and Conquer

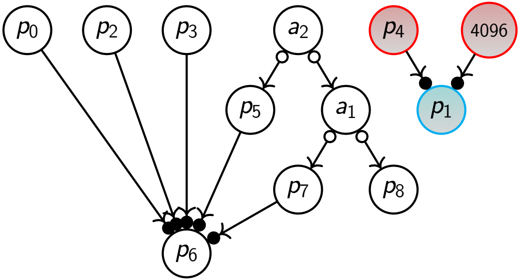


$$\begin{array}{ccccccccccccc} 2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant & 0 \\ & & & & & & & & p_8 & - & \color{blue}{p_1} & \geqslant & 0 \end{array}$$

$$\Updownarrow$$

Running example

Project and Conquer



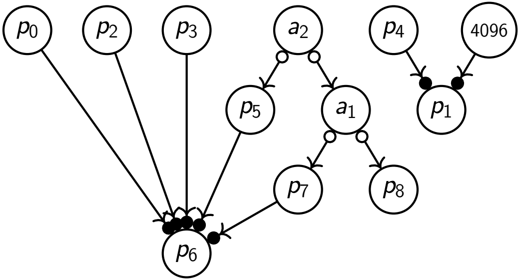
$$\begin{array}{ccccccccccccl} 2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant 0 \\ & & & & & & & & p_8 & - & \color{blue}{p_1} & \geqslant 0 \end{array}$$

$$\Updownarrow$$

$$\begin{array}{ccccccccccccl} 2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant 0 \\ & & & & & & & & p_8 & - & \color{red}{(p_4 + 4096)} & \geqslant 0 \end{array}$$

Running example

Project and Conquer



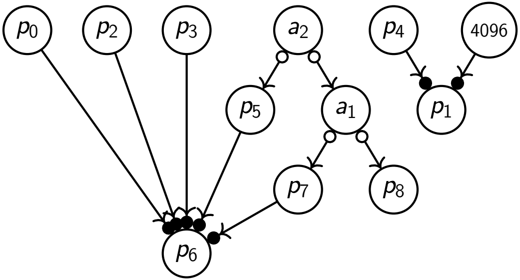
$$\begin{array}{lllllllllll}
2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_3 & - & p_5 & \geqslant 0 \\
1\,p_8 & - & p_1 & & & & & & & \geqslant 0
\end{array}$$

$$\Updownarrow$$

$$\begin{array}{lllllllllllll}
2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & & p_3 & - & p_5 & \geqslant 0 \\
 & & & & & & p_8 & - & p_4 & - & 4096 & \geqslant 0
\end{array}$$

Running example

Project and Conquer

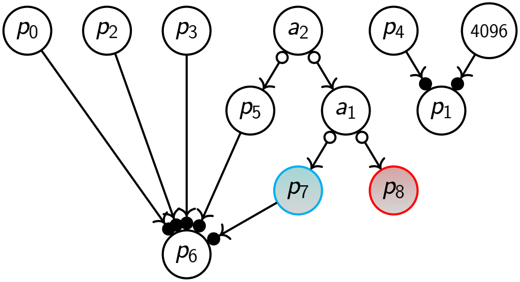


$$
\begin{array}{lllllllllllll}
2\,p_7 & + & 2\,p_8 & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant 0 \\
0\,p_7 & + & 1\,p_8 & - & p_4 & - & 4096 & & & & & \geqslant 0
\end{array}
$$

$$\Updownarrow$$

Running example

Project and Conquer



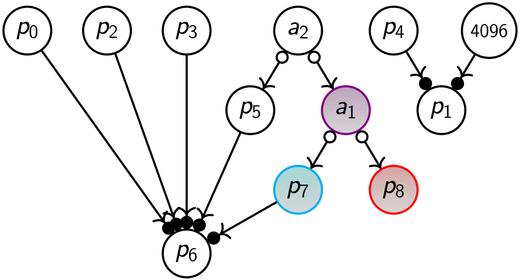
$$
\begin{array}{llllllllll}
\color{blue}{2\,p_7} & + & \color{red}{2\,p_8} & - & p_0 & - & p_2 & - & p_3 & - & p_5 & \geqslant 0 \\
\color{blue}{0\,p_7} & + & \color{red}{1\,p_8} & - & p_4 & - & 4096 & & & & & \geqslant 0
\end{array}
$$

$$\Updownarrow$$

polarized: p_8 variable with the highest coefficient in both literals

Running example

Project and Conquer



$$\begin{array}{l}
{\color{blue}2\,p_7} + {\color{red}2\,p_8} - p_0 - p_2 - p_3 - p_5 \geqslant 0 \\
{\color{blue}0\,p_7} + {\color{red}1\,p_8} - p_4 - 4096 \geqslant 0
\end{array}$$

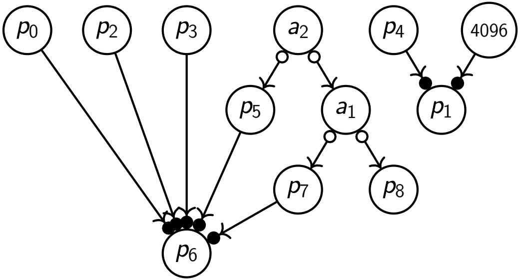
$$\Updownarrow$$

$$\begin{array}{l}
{\color{purple}2\,a_1} - p_0 - p_2 - p_3 - p_5 \geqslant 0 \\
{\color{purple}1\,a_1} - p_4 - 4096 \geqslant 0
\end{array}$$

polarized: p_8 variable with the highest coefficient in both literals

Running example

Project and Conquer

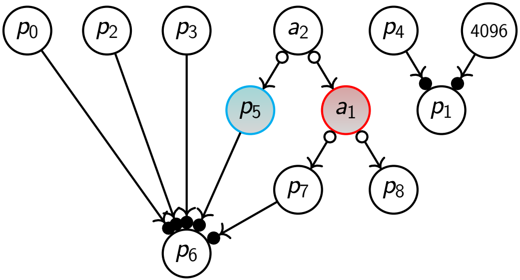


$$
\begin{array}{llllllllll}
2\,a_1 & - & 1\,p_5 & - & p_0 & - & p_2 & - & p_3 & \geqslant 0 \\
1\,a_1 & + & 0\,p_5 & - & p_4 & - & 4096 & & & \geqslant 0
\end{array}
$$

$$\Updownarrow$$

Running example

Project and Conquer



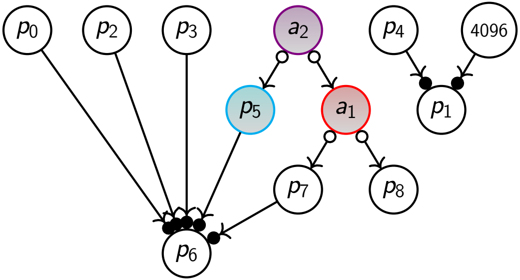
$$
\begin{array}{llllllllllll}
{\color{red}2\,a_1} & {\color{blue}-} & {\color{blue}1\,p_5} & - & p_0 & - & p_2 & - & p_3 & \geqslant 0 \\
{\color{red}1\,a_1} & {\color{blue}+} & {\color{blue}0\,p_5} & - & p_4 & - & 4096 & & & \geqslant 0
\end{array}
$$

$$\Updownarrow$$

polarized: a_1 variable with the highest coefficient in both literals

Running example

Project and Conquer



$$\begin{array}{lllllllllll} 2\,a_1 & - & 1\,p_5 & - & p_0 & - & p_2 & - & p_3 & \geqslant 0 \\ 1\,a_1 & + & 0\,p_5 & - & p_4 & - & 4096 & & & \geqslant 0 \end{array}$$

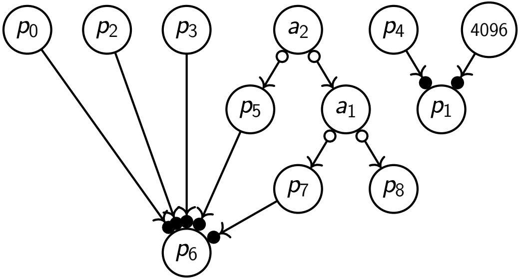
$$\Updownarrow$$

$$\begin{array}{lllllllll} 2\,a_2 & - & p_0 & - & p_2 & - & p_3 & \geqslant 0 \\ 1\,a_2 & - & p_4 & - & 4096 & & & \geqslant 0 \end{array}$$

polarized: a_1 variable with the highest coefficient in both literals

Running example

Project and Conquer



$$
\begin{array}{rcrcrl}
3\,p_7 & + & 2\,p_8 & - & p_6 & \geqslant 0 \\
 & & p_8 & - & p_1 & \geqslant 0
\end{array}
$$

$$\Updownarrow$$

$$
\begin{array}{rcrcrcrl}
2\,a_2 & - & p_0 & - & p_2 & - & p_3 & \geqslant 0 \\
a_2 & - & p_4 & - & 4096 & & & \geqslant 0
\end{array}
$$

$$F_2 \triangleq (2a_2 \geqslant p_0 + p_2 + p_3) \wedge (a_2 \geqslant p_4 + 4096)$$

If not polarized?

Project and Conquer

- **under-approximation**: If $m_2 \models F_2$ then $\exists m_1$ s.t. $m_1 \equiv_E m_2$ and $m_1 \models F_1$
- **over-approximation**: If $m_1 \models F_1$ then $\exists m_2$ s.t. $m_1 \equiv_E m_2$ and $m_2 \models F_2$

If not polarized?

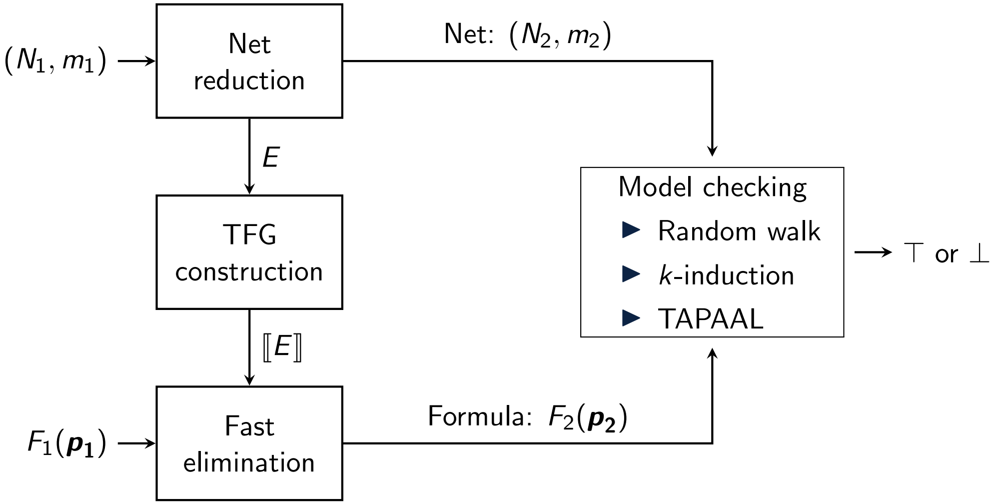
Project and Conquer

- **under-approximation**: If $m_2 \models F_2$ then $\exists m_1$ s.t. $m_1 \equiv_E m_2$ and $m_1 \models F_1$
- **over-approximation**: If $m_1 \models F_1$ then $\exists m_2$ s.t. $m_1 \equiv_E m_2$ and $m_2 \models F_2$

In practice, 80% of the formulas are polarized!

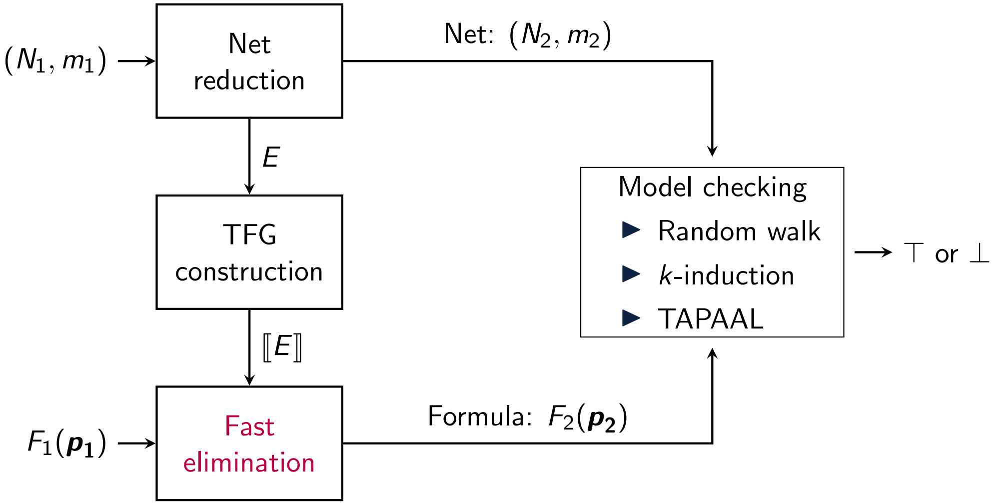
Workflow

Project and Conquer

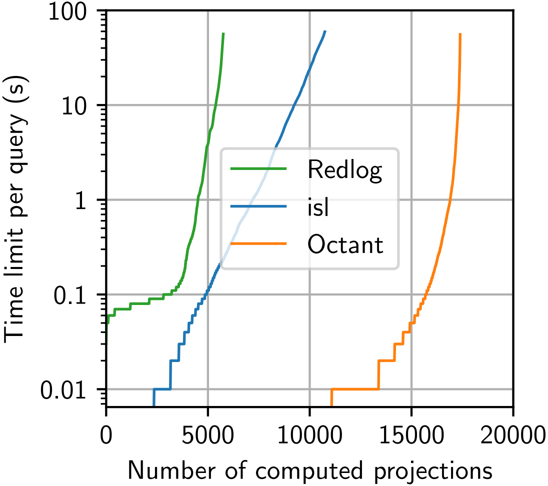


Workflow

Project and Conquer



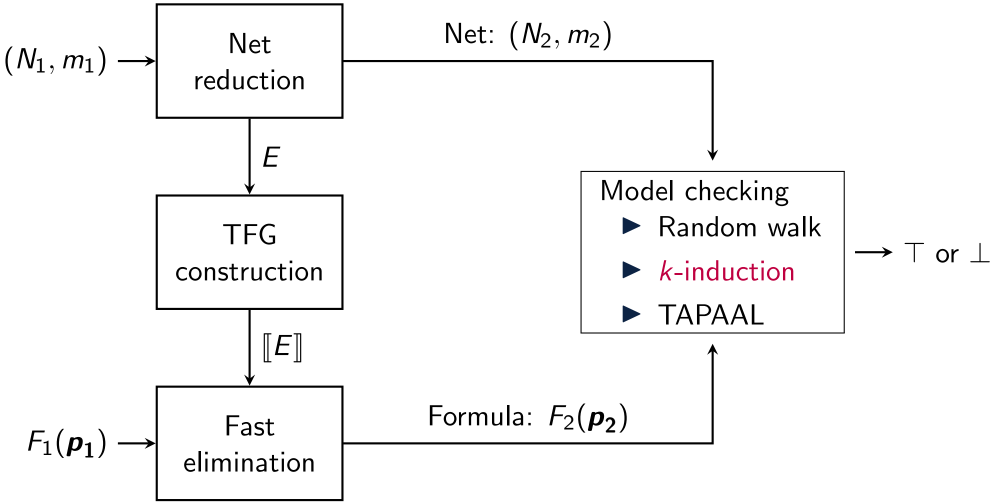
Performance of fast elimination



Octant:	99.5%
isl:	61%
Redlog:	33%

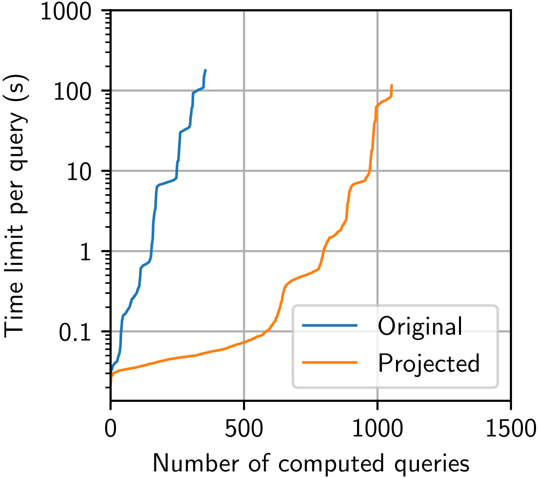
Workflow

Project and Conquer



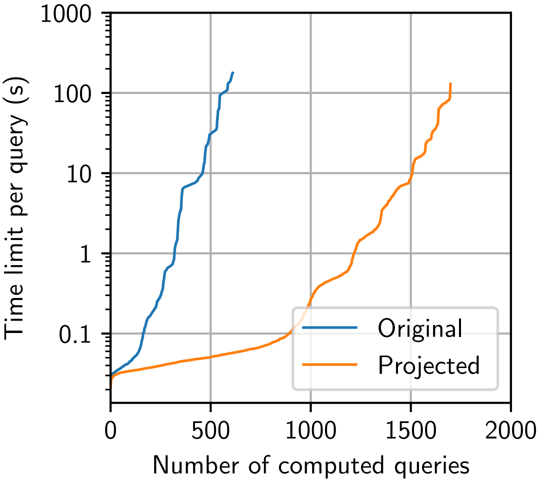
Gains with k-induction: $50\% \leqslant$ reduction ratio $\leqslant 100\%$

Project and Conquer



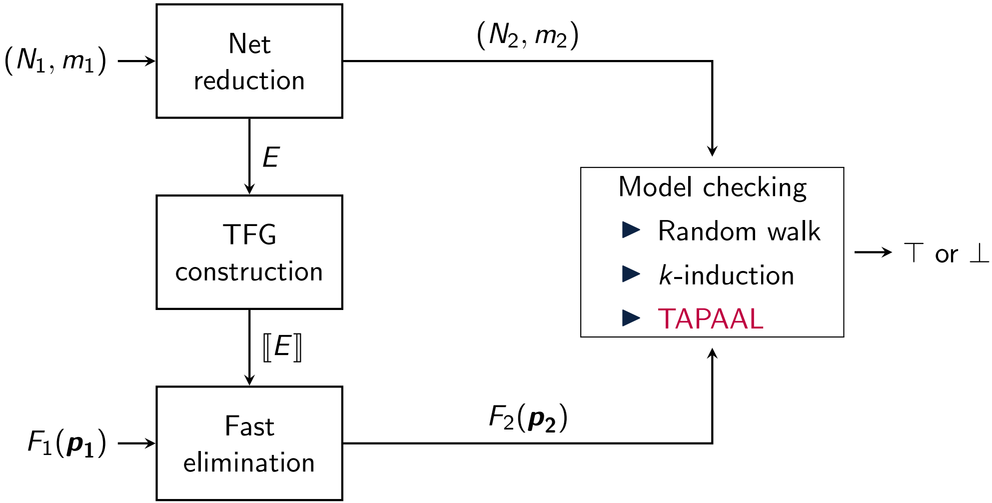
Gains with k-induction: $1\% \leqslant$ reduction ratio $\leqslant 50\%$

Project and Conquer



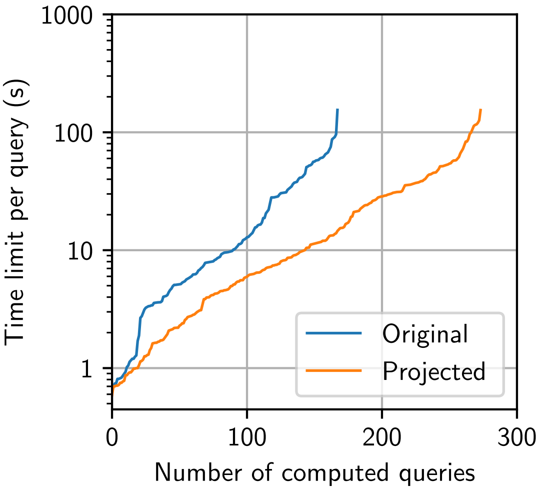
Workflow

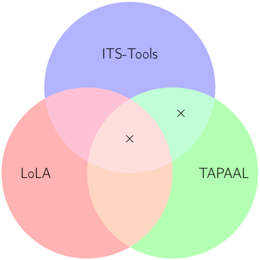
Project and Conquer



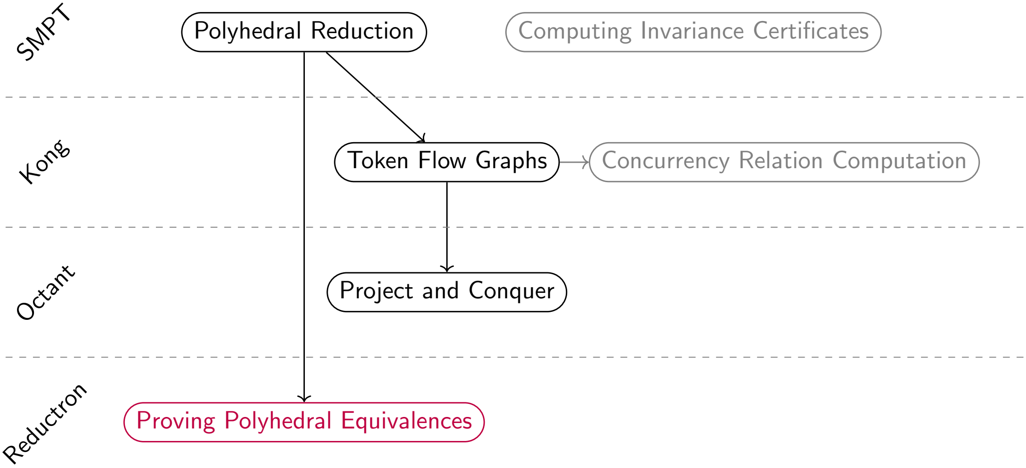
Gains with TAPAAL: challenging queries

Project and Conquer





Outline


SMPT
Polyhedral Reduction
Computing Invariance Certificates
Kong
Token Flow Graphs
Concurrency Relation Computation
Octant
Project and Conquer
Reductron
Proving Polyhedral Equivalences

Undecidability

Proving Polyhedral Equivalence

Theorem

The problem of checking a statement $(N_1, m_1) \equiv_E (N_2, m_2)$ is undecidable.

Theorem

The problem of checking a statement $(N_1, m_1) \equiv_E (N_2, m_2)$ is undecidable.

Proof.

- When $E \triangleq \mathrm{True}$: equivalent to the marking equivalence problem
- Undecidable from [Hack 76]

$\square$

Challenges and proposal

Challenges:

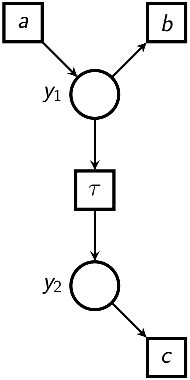
- More general notion of equivalence with a complete procedure
- Presburger sets of initial markings C_1, C_2

Proposal:

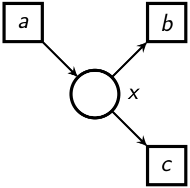
- Parametric polyhedral equivalence, $(N_1, C_1) \approxeq_E (N_2, C_2)$
- SMT constraints that ensure the equivalence

Parametric nets

Proving Polyhedral Equivalence

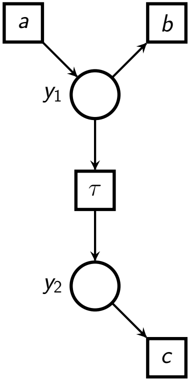


$$\approxeq_{x = y_1 + y_2}$$

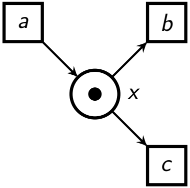


Parametric nets

Proving Polyhedral Equivalence



$\approxeq_{x = y_1 + y_2}$

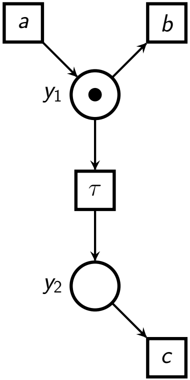


$\sigma_1 \triangleq$

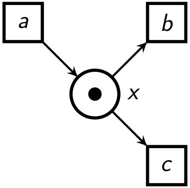
$\sigma_2 \triangleq a$

Parametric nets

Proving Polyhedral Equivalence



$$\approxeq x = y_1 + y_2$$

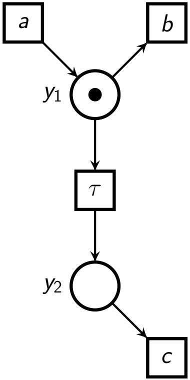


$\sigma_1 \triangleq a$

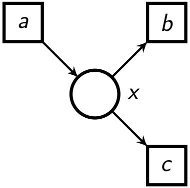
$\sigma_2 \triangleq a$

Parametric nets

Proving Polyhedral Equivalence



$$\approxeq_{x = y_1 + y_2}$$

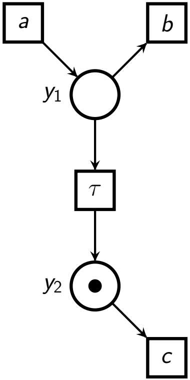


$\sigma_1 \triangleq a$

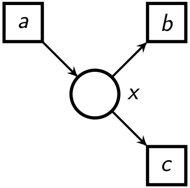
$\sigma_2 \triangleq a{\cdot}c$

Parametric nets

Proving Polyhedral Equivalence



$$\approxeq_{x = y_1 + y_2}$$

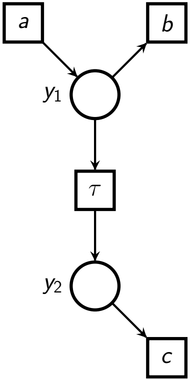


$\sigma_1 \triangleq a$

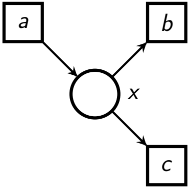
$\sigma_2 \triangleq a{\cdot}c$

Parametric nets

Proving Polyhedral Equivalence



$$\approxeq_{x = y_1 + y_2}$$

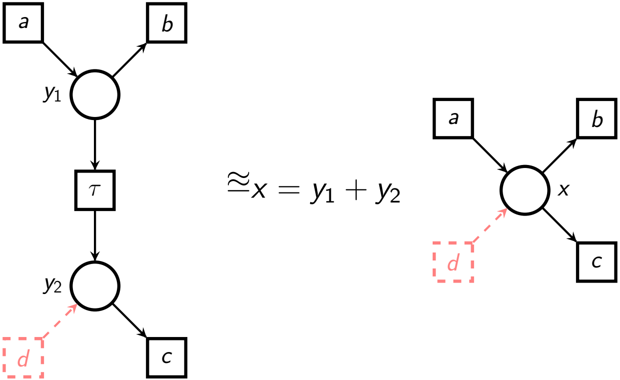


$\sigma_1 \triangleq a{\cdot}c$

$\sigma_2 \triangleq a{\cdot}c$

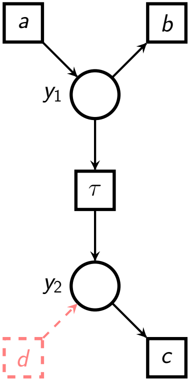
Parametric nets

Proving Polyhedral Equivalence

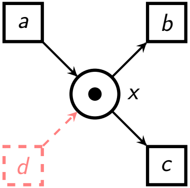


Parametric nets

Proving Polyhedral Equivalence



$$\approxeq_{x = y_1 + y_2}$$

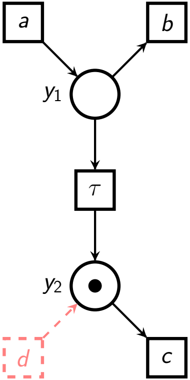


$\sigma_1 \triangleq$

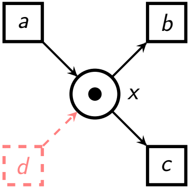
$\sigma_2 \triangleq d$

Parametric nets

Proving Polyhedral Equivalence



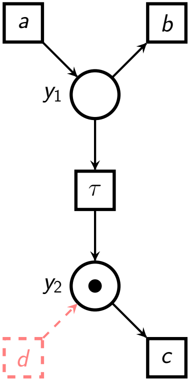
$\approx_{x = y_1 + y_2}$



$\sigma_1 \triangleq d$ $\qquad$ $\sigma_2 \triangleq d$

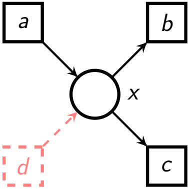
Parametric nets

Proving Polyhedral Equivalence



$\sigma_1 \triangleq d$

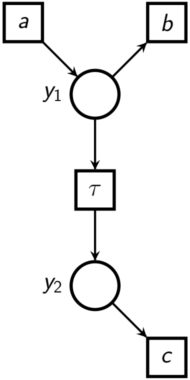
$$\approxeq_{x = y_1 + y_2}$$



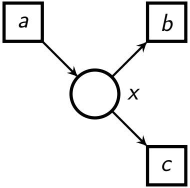
$\sigma_2 \triangleq d \cdot b$

Parametric nets

Proving Polyhedral Equivalence



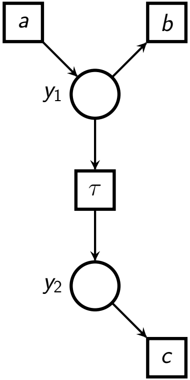
$\approxeq_{x = y_1 + y_2}$



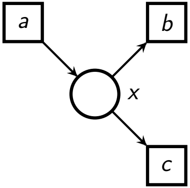
τ transitions may be irreversible choices

Parametric nets

Proving Polyhedral Equivalence



$$\approxeq_{x = y_1 + y_2}$$



$$C_1 \triangleq (y_2 = 0) \qquad C_2 \triangleq \mathrm{True}$$

Equivalence rule $[\textsc{concat}]$, $(N_1, C_1) \approxeq_E (N_2, C_2)$

Silent state-spaces

Proving Polyhedral Equivalence

To prove $(N_1, C_1) \approxeq_E (N_2, C_2)$ we need to express $m \overset{\epsilon}{\Rightarrow} m'$ with $m \models C_1$ or $m \models C_2$

Definition (Coherent net (N,C))

If $m \overset{\sigma}{\Rightarrow} m'$ with $m \in C$ then $\exists m'' \in C \;.\; m \overset{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \overset{\epsilon}{\Rightarrow} m'$.

Silent state-spaces

Proving Polyhedral Equivalence

To prove $(N_1, C_1) \approxeq_E (N_2, C_2)$ we need to express $m \overset{\epsilon}{\Rightarrow} m'$ with $m \models C_1$ or $m \models C_2$

Definition (Coherent net (N,C))

If $m \overset{\sigma}{\Rightarrow} m'$ with $m \in C$ then $\exists m'' \in C \;.\; m \overset{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \overset{\epsilon}{\Rightarrow} m'$.

A Presburger predicate, say τ_C^* such that

$$R_\tau(N, C) = \{m' \mid m' \models \exists \boldsymbol{x} \;.\; C(\boldsymbol{x}) \wedge \tau_C^*(\boldsymbol{x}, \boldsymbol{x'})\}$$

Silent state-spaces

Proving Polyhedral Equivalence

To prove $(N_1, C_1) \approxeq_E (N_2, C_2)$ we need to express $m \stackrel{\epsilon}{\Rightarrow} m'$ with $m \models C_1$ or $m \models C_2$

Definition (Coherent net (N,C))

If $m \stackrel{\sigma}{\Rightarrow} m'$ with $m \in C$ then $\exists m'' \in C \;.\; m \stackrel{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \stackrel{\epsilon}{\Rightarrow} m'$.

A Presburger predicate, say τ_C^* such that

$$R_\tau(N, C) = \{m' \mid m' \models \exists \boldsymbol{x} \;.\; C(\boldsymbol{x}) \wedge \tau_C^*(\boldsymbol{x}, \boldsymbol{x'})\}$$

Theorem

Given a parametric E-abstraction equivalence $(N_1, C_1) \approxeq_E (N_2, C_2)$, the silent reachability sets $R_\tau(N_1, C_1)$ and $R_\tau(N_2, C_2)$ are Presburger-definable.

Theorem (Leroux, 2013)

For every VASS V, for every Presburger set C_{in} of configurations, the reachability set $\mathrm{ReachV}(C_{in})$ is Presburger if, and only if, V is flattable from C_{in}.

Theorem (Leroux, 2013)

For every VASS V, for every Presburger set C_{in} of configurations, the reachability set $\mathrm{ReachV}(C_{in})$ is Presburger if, and only if, V is flattable from C_{in}.

If candidate correct: we have methods to compute τ_C^* (thanks FAST)

Decidability

Proving Polyhedral Equivalence

Theorem
The problem of checking a statement $(N_1, C_1) \approxeq_E (N_2, C_2)$ *is decidable.*

Theorem

The problem of checking a statement $(N_1, C_1) \approxeq_E (N_2, C_2)$ *is decidable.*

Proof.

- $(N_1, C_1) \approxeq_E (N_2, C_2)$ holds iff $\models$ (Core 0) $\ldots$ $\models$ (Core 3)

$\square$

Theorem

The problem of checking a statement $(N_1, C_1) \approxeq_E (N_2, C_2)$ is decidable.

Proof.

- $(N_1, C_1) \approxeq_E (N_2, C_2)$ holds iff $\models$ (Core 0) $\ldots \models$ (Core 3)
- Presburger arithmetic is decidable

$\square$

Theorem

The problem of checking a statement $(N_1, C_1) \approxeq_E (N_2, C_2)$ is decidable.

Proof.

- $(N_1, C_1) \approxeq_E (N_2, C_2)$ holds iff $\models$ (Core 0) $\ldots \models$ (Core 3)
- Presburger arithmetic is decidable
- τ_C^* can be computed using FAST if nets are flat

$\square$

Theorem
The problem of checking a statement $(N_1, C_1) \approxeq_E (N_2, C_2)$ *is decidable.*

Proof.

- $(N_1, C_1) \approxeq_E (N_2, C_2)$ holds iff $\models$ (Core 0) $\ldots$ $\models$ (Core 3)
- Presburger arithmetic is decidable
- τ_C^* can be computed using FAST if nets are flat
- Flat $\leftrightarrow$ Presburger-definable (decidable [Hauschildt 90][Lambert 94])

$\square$

Parametric equivalence instantiation

Theorem (Parametric E-abstraction Instantiation)

Assume $(N_1, C_1) \approxeq_E (N_2, C_2)$ is a parametric E-abstraction. Then,

$$m_1 \equiv_E m_2 \wedge m_1 \models C_1 \wedge m_2 \models C_2 \implies (N_1, m_1) \equiv_E (N_2, m_2)$$

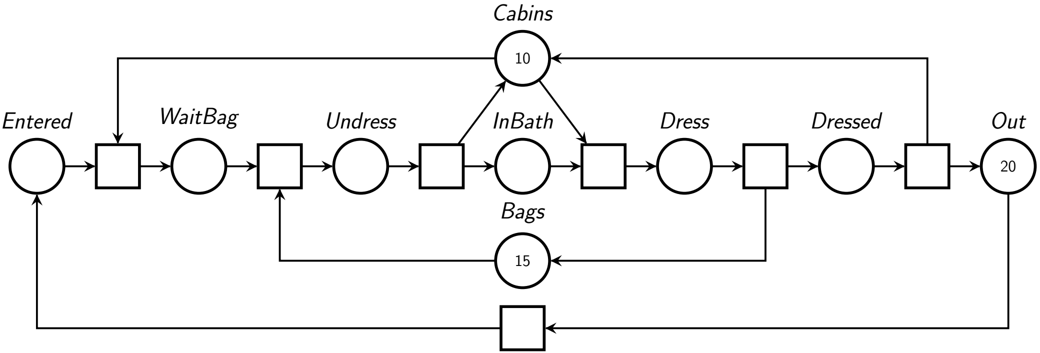
Performance evaluation

Proving Polyhedral Equivalence

- Proved our rules in less than 1 s ($[\text{RED}]$, $[\text{AGG}]$, $[\text{CONCAT}]$, etc.)
- Tested unsound rules $\rightarrow$ return which constraint failed

Performance evaluation: SwimmingPool

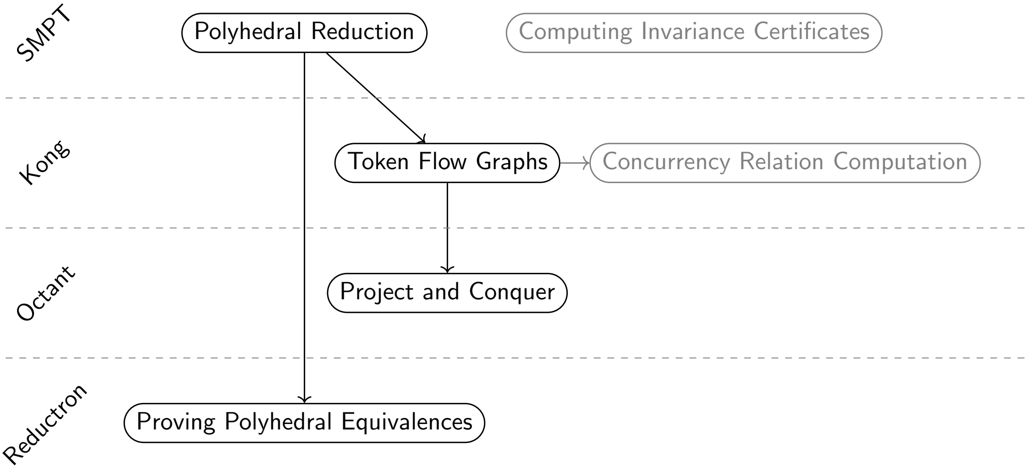
Proving Polyhedral Equivalence



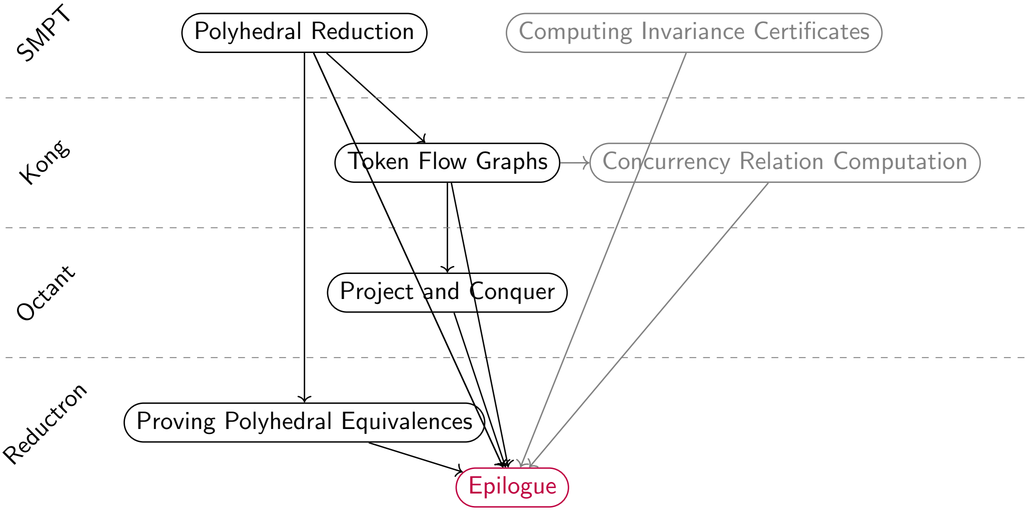
$$E \triangleq \begin{cases} Cabins + Dress + Dressed + Undress + WaitBag = 10 \\ Dress + Dressed + Entered + InBath + Out + Undress + WaitBag = 20 \\ Bags + Dress + InBath + Undress = 15 \end{cases}$$

Proving time: 11 s

Outline


SMPT
Polyhedral Reduction
Computing Invariance Certificates
Kong
Token Flow Graphs
Concurrency Relation Computation
Octant
Project and Conquer
Reductron
Proving Polyhedral Equivalences

Outline


SMPT
Kong
Octant
Reductron
Polyhedral Reduction
Computing Invariance Certificates
Token Flow Graphs
Concurrency Relation Computation
Project and Conquer
Proving Polyhedral Equivalences
Epilogue

Open science

- Making **papers accessible**
  - HAL, arXiv

Creative Commons

Open science

- Making **papers accessible**
  - HAL, arXiv
- Experimenting on **accessible benchmarks**
  - Model Checking Contest

Creative Commons

Open science

- Making **papers accessible**
  - HAL, arXiv
- Experimenting on **accessible benchmarks**
  - Model Checking Contest
- Producing **available tools** and **artifacts**
  - Open source tools available on GitHub
  - Conference artifacts: TACAS, FM, VMCAI
  - Artifact accompanying my manuscript

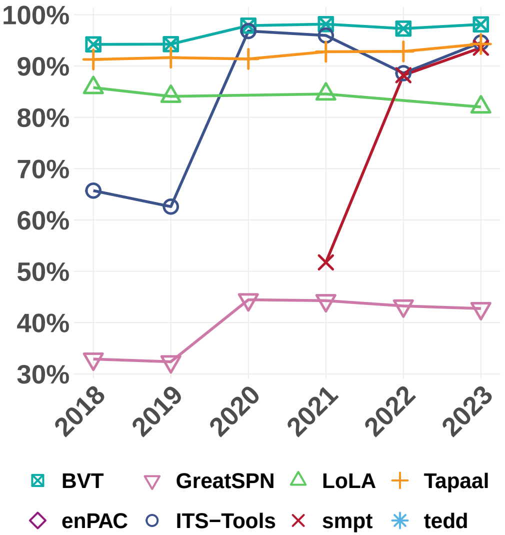
Creative Commons

Open science

- Making **papers accessible**
  - HAL, arXiv
- Experimenting on **accessible benchmarks**
  - Model Checking Contest
- Producing **available tools** and **artifacts**
  - Open source tools available on GitHub
  - Conference artifacts: TACAS, FM, VMCAI
  - Artifact accompanying my manuscript
- Participating in **competitions**
  - Model Checking Contest (2021 – 2023)

Creative Commons

Model Checking Contest (2021 – 2023)

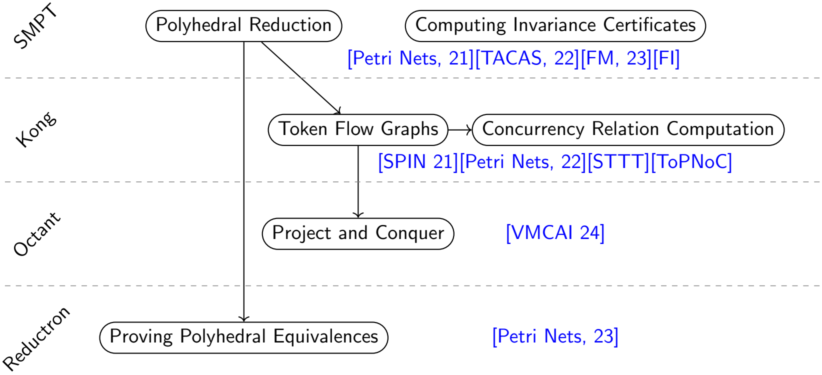


2021: BMC & PDR (coverability)

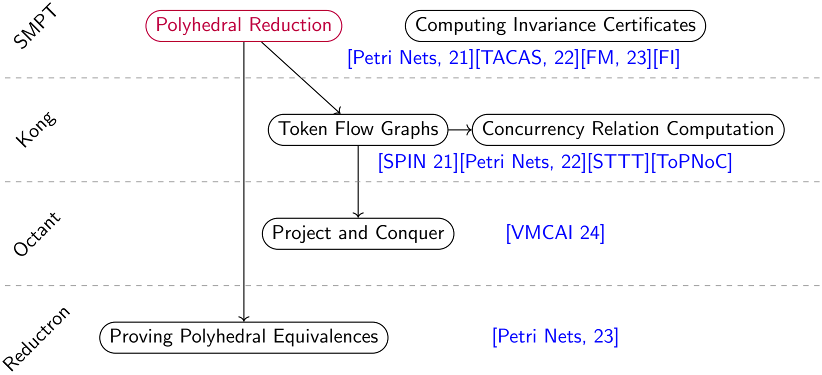
2022: Added standard methods

2023: Projection (+5.5%)

Contributions

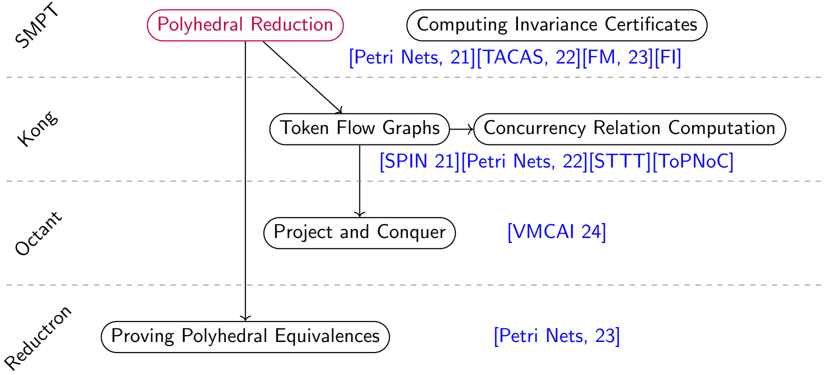


Contributions



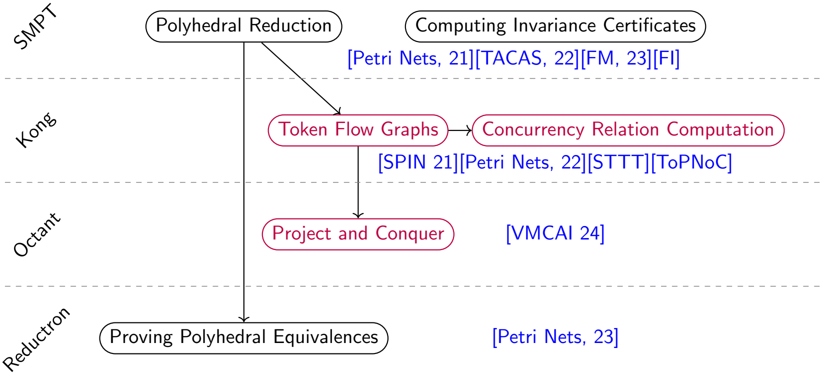
- We use a set of **simple** reductions, which are surprisingly **efficient** to reduce the net size when used together.

Contributions



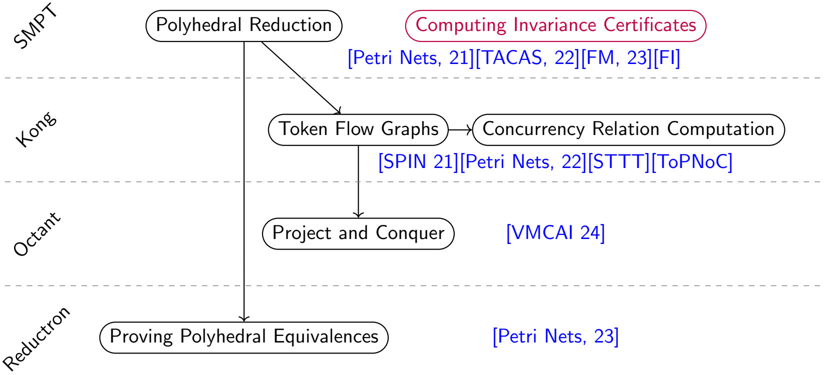
- Reductions generate linear equations which **characterize the state space** (partially or totally).

Contributions



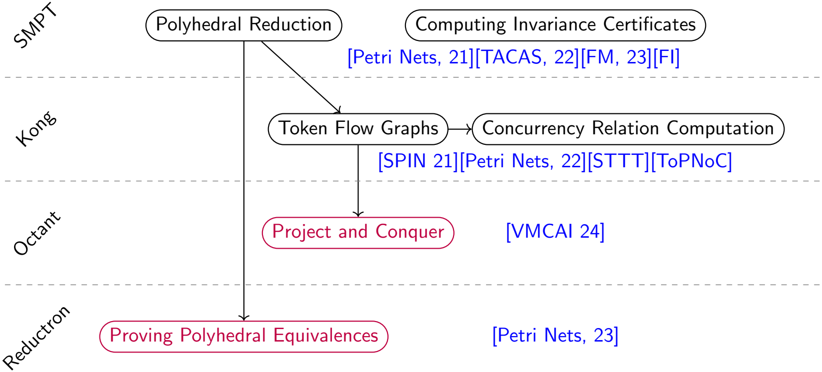
- We defined methods, and **data structures**, to **transfer problems** between the initial and the reduced net. For the **concurrency relation computation**, complexity is **linear** in the size of the output.

Contributions



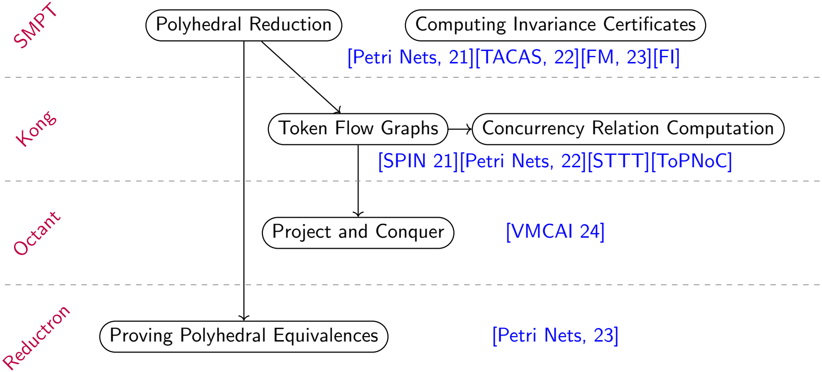
- We developed new SMT-based methods that works as well on bounded as unbounded nets, and that provides **certificate of invariance**.

Contributions



- **Unexpected**: quantifier elimination and automated proving.

Contributions

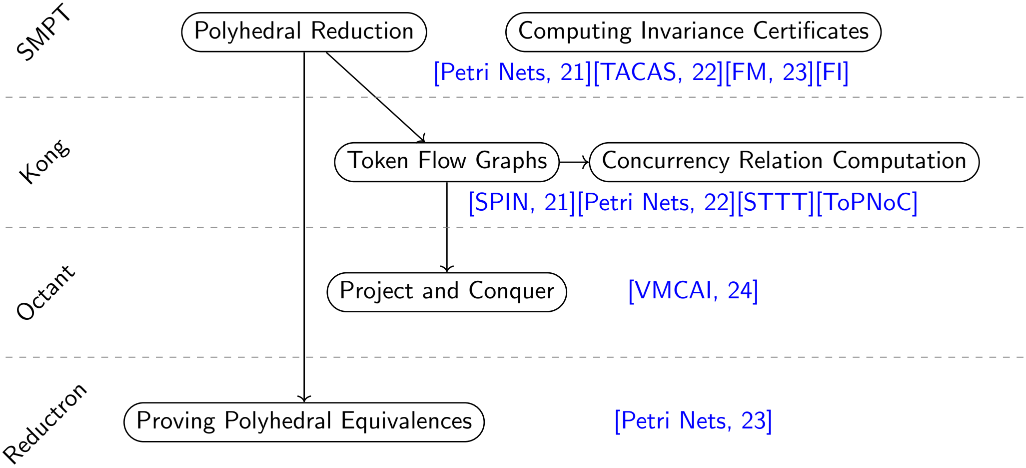


- A toolbox composed of **four open-source tools**

Perspectives

- **Reachability problem**
  - Easy at a first glance, but has picked the interest of researchers for decades
- Plenty of room to develop **new semi-procedures** and **improve existing** ones
- SMT-solvers are too general
  - Specific solvers **taking into account the underlying model**
  - Continue to explore relation with **Presburger arithmetic**

Questions?


SMPT
Polyhedral Reduction
Computing Invariance Certificates
[Petri Nets, 21][TACAS, 22][FM, 23][FI]
Kong
Token Flow Graphs
Concurrency Relation Computation
[SPIN, 21][Petri Nets, 22][STTT][ToPNoC]
Octant
Project and Conquer
[VMCAI, 24]
Reductron
Proving Polyhedral Equivalences
[Petri Nets, 23]