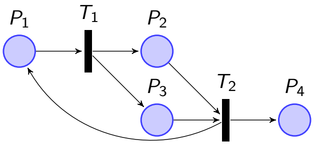
Linear algebra + Petri nets

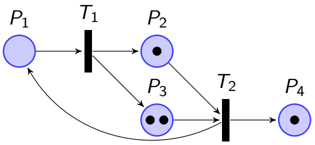
Piotr Hofman
University of Warsaw

Petri Nets.



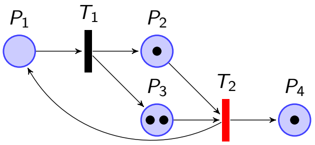
- Places.
- Transitions.

Petri Nets.



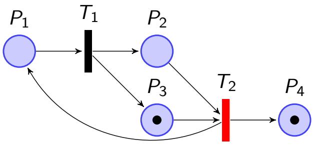
- Places.
- Transitions.
- Tokens, a Marking.

Petri Nets.



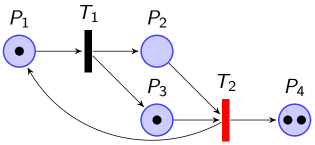
- Places.
- Transitions.
- Tokens, a Marking.
- Firing a transition.

Petri Nets.



- Places.
- Transitions.
- Tokens, a Marking.
- Firing a transition.

Petri Nets.



- Places.
- Transitions.
- Tokens, a Marking.
- Firing a transition.

Questions and tools.

We focus on analysis of systems modelled with Petri nets.

Most important questions:

1. Place coverability,
2. Reachability,
3. Liveness,
4. Death of a transition,
5. Deadlock-freeness.

Most important tools:

1. Coverability: ExpSpace complete,
2. Boundedness: ExpSpace complete,
3. Reachability: at least ExpSpace Hard.

Two solutions:

Do not try to be precise (approximations).

1. Place invariant.
2. State equation.
3. Continuous reachability.
4. Traps and siphons.

Do not try to be general (sub-classes).

1. Free-choice Petri Nets.
2. Conflict free Petri nets.
3. One counter systems.
4. 2-dimensional VASS.
5. Flat systems.

Linear algebra

Integer programming.

Input: An integer matrix M and a vector $\boldsymbol{y}$.

Question: If there is a vector $\boldsymbol{x} \in \mathbb{N}^d$ such that

$$M \cdot \boldsymbol{x} = \boldsymbol{y}?$$

Theorem

The integer programming problem is NP-complete.

Linear algebra.

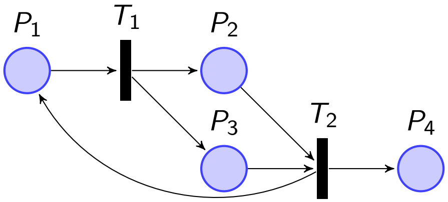
Linear programming.

Input:	An integer matrix M and a vector $\boldsymbol{y}$.
Question:	If there is a vector $\boldsymbol{x} \in \mathbb{Q}^d_{\geqslant 0}$ such that $$M \cdot \boldsymbol{x} = \boldsymbol{y}?$$

Theorem

The linear programming problem is P-complete.

Description of the net, three matrices.



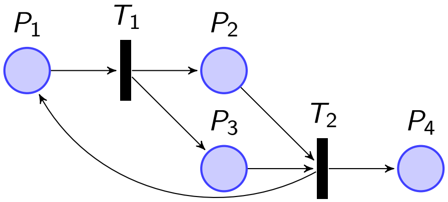
$$Pre(\mathcal{N}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$Post(\mathcal{N}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Delta = Post(\mathcal{N}) - Pre(\mathcal{N})$$

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}$$

Description of the net, three matrices.



$$Pre(\mathcal{N}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$Post(\mathcal{N}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{0}[i] = 0 \text{ for all } i$$

$$\Delta = Post(\mathcal{N}) - Pre(\mathcal{N})$$

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{1}_{\boldsymbol{p}}[i] = \begin{cases} 1 & \text{if } p = i \\ 0 & \text{otherwise} \end{cases}$$

State equation.

Let $\mathit{Reach}(\mathcal{N}, \mathfrak{i})$ be a set of configurations reachable from $\mathfrak{i}$ in $\mathcal{N}$.

State equation.

Let $Reach(\mathcal{N}, \mathfrak{i})$ be a set of configurations reachable from $\mathfrak{i}$ in $\mathcal{N}$.

Let $L_{\mathbb{N}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{N}^d}\ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

State equation.

Let $\mathit{Reach}(\mathcal{N}, \mathfrak{i})$ be a set of configurations reachable from $\mathfrak{i}$ in $\mathcal{N}$.

Hard to describe.

Let $L_{\mathbb{N}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{N}^d} \; \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

Easier to describe (NP-complete).

State equation.

Let $Reach(\mathcal{N}, \mathfrak{i})$ be a set of configurations reachable from $\mathfrak{i}$ in $\mathcal{N}$.

Hard to describe.

Let $L_{\mathbb{N}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{N}^d}\ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

Easier to describe (NP-complete).

Let $L_{\mathbb{Z}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{Z}^d}\ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

State equation.

Let $Reach(\mathcal{N}, \mathfrak{i})$ be a set of configurations reachable from $\mathfrak{i}$ in $\mathcal{N}$.

Hard to describe.

Let $L_{\mathbb{N}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{N}^d} \ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

Easier to describe (NP-complete).

Let $L_{\mathbb{Z}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{Z}^d} \ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

Easy to describe (PTime).

State equation.

Let $Reach(\mathcal{N}, \mathfrak{i})$ be a set of configurations reachable from $\mathfrak{i}$ in $\mathcal{N}$.

Hard to describe.

Let $L_{\mathbb{N}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{N}^d} \ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

Easier to describe (NP-complete).

Let $L_{\mathbb{Z}}RS(\mathcal{N}, \mathfrak{i}) = \{\boldsymbol{y} : \exists_{\boldsymbol{x} \in \mathbb{Z}^d} \ \Delta \cdot \boldsymbol{x} = \boldsymbol{y} - \mathfrak{i}\}$.

Easy to describe (PTime).

Lemma

$Reach(\mathcal{N}, \mathfrak{i}) \subseteq L_{\mathbb{N}}RS(\mathcal{N}, \mathfrak{i}) \subseteq L_{\mathbb{Z}}RS(\mathcal{N}, \mathfrak{i})$.

An application.

Algorithm 1 for reachability.

Start from the initial configuration $\mathfrak{i}$ and exhaustively build a graph of reachable configurations adding nodes one by one.

- if you find $\mathfrak{f}$ then return 1;
- if you can not visit any new configuration then return 0;
- if you run out of memory then return I don't know.

An application.

Algorithm 1 for reachability.

Start from the initial configuration $\mathfrak{i}$ and exhaustively build a graph of reachable configurations adding nodes one by one.

- if you find $\mathfrak{f}$ then return 1;
- if you can not visit any new configuration then return 0;
- if you run out of memory then return I don't know.

Algorithm 2 for reachability.

Start from the initial configuration $\mathfrak{i}$ and exhaustively build a graph of reachable configurations adding nodes one by one; but whenever you want to add a new node $\boldsymbol{x}$ to the graph you check if $\mathfrak{f} \in L_{\mathbb{N}}SR(\mathcal{N}, \boldsymbol{x})$. You add the node if and only if the answer is yes.

- if you find $\mathfrak{f}$ then return 1;
- if you can not add any new node then return 0;
- if you run out of memory then return "I don't know".

P-flows

$\boldsymbol{y}$ is called a P-flow iff $\boldsymbol{y} \cdot \Delta = 0$. If $\boldsymbol{y} \geqslant 0$ then we call it P-semiflow.

P-flows

$\boldsymbol{y}$ is called a P-flow iff $\boldsymbol{y} \cdot \Delta = 0$. If $\boldsymbol{y} \geqslant 0$ then we call it P-semiflow.

Lemma

If $\mathfrak{f}$ is reachable from $\mathfrak{i}$ then $\boldsymbol{y} \cdot \mathfrak{f} = \boldsymbol{y} \cdot \mathfrak{i}$.

P-flows

$\boldsymbol{y}$ is called a P-flow iff $\boldsymbol{y} \cdot \Delta = 0$. If $\boldsymbol{y} \geqslant 0$ then we call it P-semiflow.

Lemma

If $\mathfrak{f}$ is reachable from $\mathfrak{i}$ then $\boldsymbol{y} \cdot \mathfrak{f} = \boldsymbol{y} \cdot \mathfrak{i}$.

Question

How do we test a boundedness of a place using P-semiflows?

P-flows

$\boldsymbol{y}$ is called a P-flow iff $\boldsymbol{y} \cdot \Delta = 0$. If $\boldsymbol{y} \geqslant 0$ then we call it P-semiflow.

Lemma

If $\mathfrak{f}$ is reachable from $\mathfrak{i}$ then $\boldsymbol{y} \cdot \mathfrak{f} = \boldsymbol{y} \cdot \mathfrak{i}$.

Question

How do we test a boundedness of a place using P-semiflows?

Lemma

Let $\boldsymbol{y}$ be a P-semiflow of the net $\mathcal{N}$, then the number of tokens is bounded for all $1 \leqslant i \leqslant d$ such that $\boldsymbol{y}[i] > 0$.

Structural boundedness

A place p in a net $\mathcal{N}$ is structurally bounded if for every initial marking $\mathfrak{i}$ the

$$max\{\mathbf{1_p}^T \cdot \boldsymbol{m} : \boldsymbol{m} \in RS(\mathcal{N}, \mathfrak{i})\} \text{ is finite.}$$

Structural boundedness

A place p in a net $\mathcal{N}$ is structurally bounded if for every initial marking $\mathfrak{i}$ the

$$max\{\mathbf{1_p}^T \cdot \boldsymbol{m} : \boldsymbol{m} \in RS(\mathcal{N}, \mathfrak{i})\} \text{ is finite.}$$

Theorem

A following conditions are equivalent:

1. a place p in the net $\mathcal{N}$ is structurally bounded,
2. there exists $\boldsymbol{y} \geqslant \mathbf{1_p}$ such that $\boldsymbol{y} \cdot \Delta \leqslant \mathbf{0}$,
3. there is no $\boldsymbol{x} \geqslant \mathbf{0}$ such that $\Delta \cdot \boldsymbol{x} \geqslant \mathbf{1_p}$.

Proof

Theorem

A following conditions are equivalent:

1. a place p in the net $\mathcal{N}$ is structurally bounded,
2. there exists $\boldsymbol{y} \geqslant \mathbf{1}_{\boldsymbol{p}}$ such that $\boldsymbol{y} \cdot \Delta \leqslant \mathbf{0}$,
3. there is no $\boldsymbol{x} \geqslant \mathbf{0}$ such that $\Delta \cdot \boldsymbol{x} \geqslant \mathbf{1}_{\boldsymbol{p}}$.

Proof

Theorem

A following conditions are equivalent:

1. a place p in the net $\mathcal{N}$ is structurally bounded,
2. there exists $\boldsymbol{y} \geqslant \mathbf{1_p}$ such that $\boldsymbol{y} \cdot \Delta \leqslant \mathbf{0}$,
3. there is no $\boldsymbol{x} \geqslant \mathbf{0}$ such that $\Delta \cdot \boldsymbol{x} \geqslant \mathbf{1_p}$.

1. $1 \implies 3$ by $\neg 3 \implies \neg 1$

Proof

Theorem

A following conditions are equivalent:

1. a place p in the net $\mathcal{N}$ is structurally bounded,
2. there exists $\boldsymbol{y} \geqslant \mathbf{1}_{\boldsymbol{p}}$ such that $\boldsymbol{y} \cdot \Delta \leqslant \mathbf{0}$,
3. there is no $\boldsymbol{x} \geqslant \mathbf{0}$ such that $\Delta \cdot \boldsymbol{x} \geqslant \mathbf{1}_{\boldsymbol{p}}$.

1. $1 \implies 3$ by $\neg 3 \implies \neg 1$
2. $3 \implies 2$ by a theorem related to dual programs theorem called alternative theorem.

Theorem

Exactly one of the following systems of equations has a solution:

$$A\boldsymbol{x} \geqslant \boldsymbol{b}.$$

$$\begin{aligned} \boldsymbol{y} &\geqslant \mathbf{0} \\ \boldsymbol{y}^T \cdot A &= \mathbf{0} \\ \boldsymbol{y}^T \cdot \boldsymbol{b} &> 0. \end{aligned}$$

Proof

Theorem

A following conditions are equivalent:

1. a place p in the net $\mathcal{N}$ is structurally bounded,
2. there exists $\boldsymbol{y} \geqslant \mathbf{1}_{\boldsymbol{p}}$ such that $\boldsymbol{y} \cdot \Delta \leqslant \mathbf{0}$,
3. there is no $\boldsymbol{x} \geqslant \mathbf{0}$ such that $\Delta \cdot \boldsymbol{x} \geqslant \mathbf{1}_{\boldsymbol{p}}$.

1. $1 \implies 3$ by $\neg 3 \implies \neg 1$
2. $3 \implies 2$ by a theorem related to dual programs theorem called alternative theorem.

Theorem

Exactly one of the following systems of equations has a solution:

$$A\boldsymbol{x} \geqslant \boldsymbol{b}.$$

$$\begin{aligned} \boldsymbol{y} &\geqslant \mathbf{0} \\ \boldsymbol{y}^T \cdot A &= \mathbf{0} \\ \boldsymbol{y}^T \cdot \boldsymbol{b} &> 0. \end{aligned}$$

3. $2 \implies 1$ Direct.

Continuous reachability.

Linear programming + If formula.

Input:	A $r \times c$- integer matrix M and a vector $\boldsymbol{y} \in \mathbb{Z}^r$ and a set of predicates of a form $\boldsymbol{x}[i] > 0 \implies \boldsymbol{x}[j] > 0$.
Question:	If there is a vector $\boldsymbol{x} \in \mathbb{Q}^c_{\geqslant 0}$ such that $M \cdot \boldsymbol{x} = \boldsymbol{y}$ and all predicates are satisfied?

Theorem

The Linear programming + If formula problem is in PTime.

Linear programming + If formula.

Input:	A $r \times c$- integer matrix M and a vector $\boldsymbol{y} \in \mathbb{Z}^r$ and a set of predicates of a form $\boldsymbol{x}[i] > 0 \implies \boldsymbol{x}[j] > 0$.
Question:	If there is a vector $\boldsymbol{x} \in \mathbb{Q}^c_{\geqslant 0}$ such that $M \cdot \boldsymbol{x} = \boldsymbol{y}$ and all predicates are satisfied?

Theorem

The Linear programming + If formula problem is in PTime.

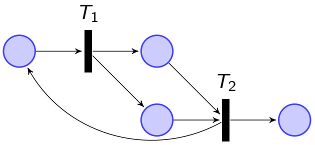
Proof

1. The set of solutions is convex.
2. If for every i there is a solution such that $\boldsymbol{x}[i] > 0$ then there is a solution such that $\boldsymbol{x}[j] > 0$ for all j.

Linear programming + If formula (the algorithm).

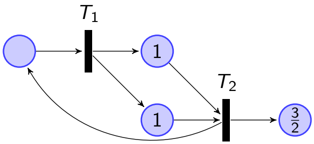
solve(Matrix Δ, Vector $\boldsymbol{y}$, set_of_implications $\mathbb{S}$, set_of_zeros $\mathbb{X}$)
{
If there is no solution $\Delta \cdot \boldsymbol{x} = \boldsymbol{y}$ in $\mathbb{Q}^{c}_{\geqslant 0}$,
where $x_i = 0$ for all $x_i \in \mathbb{X}$ then return false;
If there is a solution $\Delta \cdot \boldsymbol{x} = \boldsymbol{y}$ in $\mathbb{Q}^{c}_{\geqslant 0}$,
where $x_i = 0$ iff $x_i \in \mathbb{X}$ and $x_i > 0$ if $x_i \notin \mathbb{X}$
then return true;
Find a new coordinate x_j
which has to be equal 0 in every solution;
Add x_j to $\mathbb{X}$;
Add to $\mathbb{X}$ all x_i that has to be added due to implications;
return solve(M, $\boldsymbol{y}$, $\mathbb{S}$, $\mathbb{X}$);
}

Continuous Petri Nets.



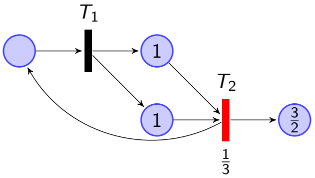
- Marking: $\mathcal{M} : \mathbb{P} \to \mathbb{Q}_{\geqslant 0}$
- Transitions: $\mathbb{T}$
- Firing a transition $\mathfrak{t} \in \mathbb{T}$ with a coefficient $a \in \mathbb{Q}_{\geqslant 0}$.

Continuous Petri Nets.



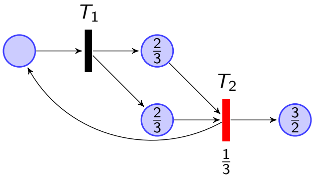
- Marking: $\mathcal{M} : \mathbb{P} \to \mathbb{Q}_{\geqslant 0}$
- Transitions: $\mathbb{T}$
- Firing a transition $\mathfrak{t} \in \mathbb{T}$ with a coefficient $a \in \mathbb{Q}_{\geqslant 0}$.

Continuous Petri Nets.



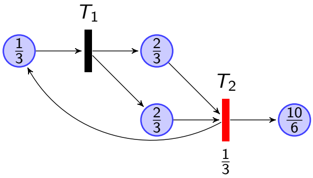
- Marking: $\mathcal{M} : \mathbb{P} \to \mathbb{Q}_{\geqslant 0}$
- Transitions: $\mathbb{T}$
- Firing a transition $\mathfrak{t} \in \mathbb{T}$ with a coefficient $a \in \mathbb{Q}_{\geqslant 0}$.

Continuous Petri Nets.



- Marking: $\mathcal{M} : \mathbb{P} \to \mathbb{Q}_{\geqslant 0}$
- Transitions: $\mathbb{T}$
- Firing a transition $\mathfrak{t} \in \mathbb{T}$ with a coefficient $a \in \mathbb{Q}_{\geqslant 0}$.

Continuous Petri Nets.



- Marking: $\mathcal{M} : \mathbb{P} \to \mathbb{Q}_{\geqslant 0}$
- Transitions: $\mathbb{T}$
- Firing a transition $\mathfrak{t} \in \mathbb{T}$ with a coefficient $a \in \mathbb{Q}_{\geqslant 0}$.

Continuous Petri Nets Reachability.

Input:	Two configurations $\mathfrak{i}$ and $\mathfrak{f}$
Question:	If there is a run form $\mathfrak{i}$ to $\mathfrak{f}$ under continuous semantics.

A simpler variant of the problem.

Suppose, that

$$\forall_i \ \left(\mathfrak{i}[i] > 0 \text{ and } \mathfrak{f}[i] > 0\right).$$

$\mathfrak{f}$ is reachable from $\mathfrak{i}$ iff

$$\mathfrak{f} - \mathfrak{i} = \Delta \cdot \boldsymbol{x} \text{ where } \boldsymbol{x} \in \mathbb{Q}^{d}_{\geqslant 0}.$$

Continuous Petri Nets Reachability.

Lemma

$\mathfrak{f}$ is reachable from $\mathfrak{i}$ if

1. $$\mathfrak{f} - \mathfrak{i} = \Delta \cdot \boldsymbol{x} \text{ where } \boldsymbol{x} \in \mathbb{Q}^d_{\geqslant 0}$$
2. $$\boldsymbol{x}[t_i] > 0 \;\text{ and }\; Pre[j, t_i] > 0 \implies \mathfrak{i}[j] > 0,$$
3. $$\boldsymbol{x}[t_i] > 0 \;\text{ and }\; Post[j, t_i] > 0 \implies \mathfrak{f}[j] > 0.$$

Continuous Petri Nets Reachability.

Lemma

$\mathfrak{f}$ is reachable from $\mathfrak{i}$ if

1. $\mathfrak{f} - \mathfrak{i} = \Delta \cdot \boldsymbol{x}$ where $\boldsymbol{x} \in \mathbb{Q}^d_{\geqslant 0}$
2. $\boldsymbol{x}[t_i] > 0$ and $Pre[j, t_i] > 0 \implies \mathfrak{i}[j] > 0$,
3. $\boldsymbol{x}[t_i] > 0$ and $Post[j, t_i] > 0 \implies \mathfrak{f}[j] > 0$.

Theorem

$\mathfrak{f}$ is reachable from $\mathfrak{i}$ iff there are two configurations $\mathfrak{i}'$ and $\mathfrak{f}'$ such that

1. there is a run form $\mathfrak{i}$ to $\mathfrak{i}'$ that is using at most d steps.
2. there is a run form $\mathfrak{f}'$ to $\mathfrak{f}$ that is using at most d steps.
3. There is a run form $\mathfrak{i}'$ to $\mathfrak{f}'$ due to Lemma.

Translation to a formula (linear + If).

Lemma

For a given Petri net $\mathcal{N}$ and two configurations $\mathfrak{i}$ and $\mathfrak{f}$ in PTime one can compute a formula (linear programming + if) such that it is satisfiable if and only if $\mathfrak{f}$ is continuously reachable from $\mathfrak{i}$ in the net $\mathcal{N}$.

We use:

Theorem

$\mathfrak{f}$ is reachable from $\mathfrak{i}$ iff there are two configurations $\mathfrak{i}'$ and $\mathfrak{f}'$ such that

1. there is a run form $\mathfrak{i}$ to $\mathfrak{i}'$ that is using at most d steps.
2. there is a run form $\mathfrak{f}'$ to $\mathfrak{f}$ that is using at most d steps.
3. There is a run form $\mathfrak{i}'$ to $\mathfrak{f}'$ due to Lemma.

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

What is the main obstacle?

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

CHALLENGE: Size of the representation of the upward-closed set may get too big.

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

CHALLENGE: Size of the representation of the upward-closed set may get too big.

How to cut the upward-closed set?

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

CHALLENGE: Size of the representation of the upward-closed set may get too big.

IDEA: Let $\boldsymbol{x} \in M\uparrow$, if there is no $\boldsymbol{y} \geqslant \boldsymbol{x}$ such that $\boldsymbol{y} \in RS(\mathcal{N}, \mathfrak{i})$ then we can throw $\boldsymbol{x}$ away.

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

CHALLENGE: Size of the representation of the upward-closed set may get too big.

IDEA: Let $\boldsymbol{x} \in M\uparrow$, if there is no $\boldsymbol{y} \geqslant \boldsymbol{x}$ such that $\boldsymbol{y} \in RS(\mathcal{N}, \mathfrak{i})$ then we can throw $\boldsymbol{x}$ away.

M. Blondin, A. Finkel, Ch. Haase, S. Haddad, 2015

SOLUTION: Let $\boldsymbol{x} \in M\uparrow$, if there is no $\boldsymbol{y} \geqslant \boldsymbol{x}$ such that $\boldsymbol{y} \in CRS(\mathcal{N}, \mathfrak{i})$ then we can throw $\boldsymbol{x}$ away.

Q-cover 2015.

IDEA: Take a backward coverability algorithm, and speed it up.

CHALLENGE: Size of the representation of the upward-closed set may get too big.

IDEA: Let $\boldsymbol{x} \in M\uparrow$, if there is no $\boldsymbol{y} \geqslant \boldsymbol{x}$ such that $\boldsymbol{y} \in RS(\mathcal{N}, \mathfrak{i})$ then we can throw $\boldsymbol{x}$ away.

M. Blondin, A. Finkel, Ch. Haase, S. Haddad, 2015

SOLUTION: Let $\boldsymbol{x} \in M\uparrow$, if there is no $\boldsymbol{y} \geqslant \boldsymbol{x}$ such that $\boldsymbol{y} \in CRS(\mathcal{N}, \mathfrak{i})$ then we can throw $\boldsymbol{x}$ away.

Thomas Geffroy, Jérôme Leroux, Grégoire Sutre, 2017

Actually, any over-approximation will work: LRS instead of CRS.

Bibliography

1. Techniques for state equation and flow invariant: https://link.springer.com/content/pdf/10.1007/3-540-65306-6_19.pdf
2. Continuous reachability:
   old paper: Estíbaliz Fraca, Serge Haddad: *Complexity Analysis of Continuous Petri Nets.* Fundam. Inform. 137(1): 1-28 (2015) (It has to be in the library)
   new paper: http://www.lsv.fr/~haase/documents/bh17.pdf

Fast Termination.

Definition (VASS- Vector addition systems with states)

VASS is a finite automaton in which transitions are labelled with vectors in $\mathbb{Z}^d$. The set of states we denote by Q and the set of transition by T. The semantics is given by a labelled transition system where:

- Configurations are pairs a state and a vector in $\mathbb{N}^d$.
- There is transition from $(p, \boldsymbol{m})$ to $(q, \boldsymbol{m'})$ if there is an automaton transition (p, q) labelled with $\boldsymbol{v}$ such that $\boldsymbol{m} + \boldsymbol{v} = \boldsymbol{m'}$.

1. $L(n)$ is the maximal length of a run from a configuration with the counters bounded by n.
2. SCC -strongly connected component in the automaton.
3. Let A be a VASS, and R its strongly connected component, by A_R we mean the VASS obtained for A by restricting the set of states to R.

Our goal is to propose algorithm that approximates a function $L(n)$.

Definition

An open half-space of $\mathbb{Q}^d$ determined by a normal vector $\boldsymbol{n} \in \mathbb{Q}^d$, where $n \neq 0$, is the set $H_{\boldsymbol{n}}$ of all $\boldsymbol{x} \in \mathbb{Q}^d$ such that $\boldsymbol{x} \cdot \boldsymbol{n} < 0$ (dot product). A closed half-space $H_{\boldsymbol{n}}$ is defined in the same way but the above inequality is non-strict.

Definition

Given a finite set of vectors $U \subseteq \mathbb{Q}^d$, we use *cone*(U) to denote the set of all vectors of the form $\sum_{\boldsymbol{u} \in U} c_{\boldsymbol{u}} \boldsymbol{u}$, where $c_{\boldsymbol{u}}$ is a non-negative rational constant for every $\boldsymbol{u} \in U$.

Hyperplane separation theorem

Let A and B be two disjoint nonempty convex subsets of $\mathbb{Q}^d$. Then there exist a nonzero vector v and a real number c such that
$\langle x, v \rangle \geq c$ and $\langle y, v \rangle \leq c$ for all $x \in A$ and $y \in B$; i.e., the hyperplane $\langle \cdot, v \rangle = c$, where v is the normal vector, separates A and B. If A and B are closed then inequality can be strict.

Lemma

Let $d \in \mathbb{N}$, and let $A = (Q, T)$ be a d-dimensional VASS. Then $L(n) \in O(n)$ iff $L_R(n) \in O(n)$ for every SCC R of Q, where $L_R(n)$ is the termination complexity of A_R.

Lemma

Let $d \in \mathbb{N}$, and let $A = (Q, T)$ be a d-dimensional VASS. Then $L(n) \in O(n)$ iff $L_R(n) \in O(n)$ for every SCC R of Q, where $L_R(n)$ is the termination complexity of A_R.

Definition

$Inc \stackrel{\text{def}}{=} \{eff(\pi) | \pi$ is a cycle in A not longer than $|Q|\}$.

Lemma

Let $A = (Q, T)$ be a d-dimensional VASS. Then one of two cases holds:

- *there exist $v_1, \ldots, v_k \in Inc$ and $b_1, \ldots, b_k \in \mathbb{N}$ such that $k \geq 1$ and $\sum_{i=1}^{k} b_i \mathbf{v_i} \geq 0$,*
- *there is an open half-space $H_{\mathbf{n}} \subset \mathbb{R}^d$ defined by $\mathbf{n} > 0$ such that $Inc \subseteq H_n$.*

Lemma

Let $A = (Q, T)$ be a d-dimensional VASS. We have the following:

- *If there is an open half-space H_s of $\mathbb{Q}^d$ such that $\boldsymbol{s} > 0$ and $Inc \subset H_s$, then $L(n) \in O(n)$.*
- *If there is a closed half-space H_s of $\mathbb{Q}^d$ such that $\boldsymbol{s} > 0$ and $Inc \subseteq H_s$, then $L(n) \in \Omega(n^2)$.*
- *If there is a vector $\boldsymbol{s} > 0$ that can be expressed as $\sum_{\boldsymbol{u} \in Inc} c_{\boldsymbol{u}} \cdot \boldsymbol{u}$ then the net has an infinite run.*

Lemma

Let $A = (Q, T)$ be a d-dimensional VASS. We have the following:

- *If there is an open half-space $H_{\mathbf{s}}$ of $\mathbb{Q}^d$ such that $\mathbf{s} > 0$ and $Inc \subset H_{\mathbf{s}}$, then $L(n) \in O(n)$.*
- *If there is a closed half-space $H_{\mathbf{s}}$ of $\mathbb{Q}^d$ such that $\mathbf{s} > 0$ and $Inc \subseteq H_{\mathbf{s}}$, then $L(n) \in \Omega(n^2)$.*
- *If there is a vector $\mathbf{s} > 0$ that can be expressed as $\sum_{\mathbf{u} \in Inc} c_{\mathbf{u}} \cdot \mathbf{u}$ then the net has an infinite run.*

Theorem

Let $d \in \mathbb{N}$. The problem whether the termination complexity of a given d-dimensional VASS is linear is solvable in time polynomial in the size of A. More precisely, the termination complexity of a VASS A is linear if and only if there exists a weighted linear ranking function for A. Moreover, the existence of a weighted linear ranking function for A can be decided in time polynomial in the size of A.

Bibliography

1. https://arxiv.org/pdf/1708.09253.pdf