The Reachability Problem for Petri Nets is Not Elementary

Wojciech Czerwiński[1], Sławomir Lasota[1], Ranko Lazić[2], Jérôme Leroux[3] and Filip Mazowiecki[3]

[1]Univ. Warsaw

[2]Univ. Warwick

[3] Univ. Bordeaux, CNRS, LaBRI

Funded by ANR project BRAVAS

The Reachability Problem for Petri Nets is Not Elementary

Wojciech Czerwiński[1], Sławomir Lasota[1], Ranko Lazić[2], Jérôme Leroux[3] and Filip Mazowiecki[3]

[1]Univ. Warsaw

[2]Univ. Warwick

[3] Univ. Bordeaux, CNRS, LaBRI

Funded by ANR project BRAVAS

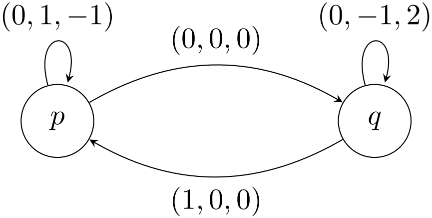
Introduction

VASS and Counter Programs

Vector addition systems with states (VASS)

(d, Q, T), where $T \subseteq Q \times \mathbb{Z}^d \times Q$

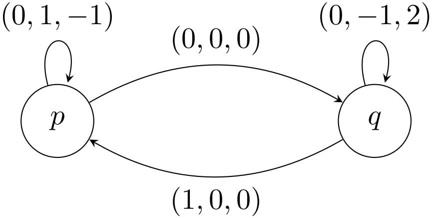
Example: $d = 3$, $Q = \{p, q\}$



Vector addition systems with states (VASS)

(d, Q, T), where $T \subseteq Q \times \mathbb{Z}^d \times Q$

Example: $d = 3$, $Q = \{p, q\}$

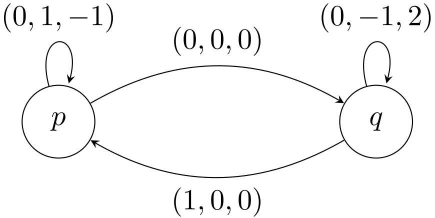


Configurations $q(\mathbf{v}) = (q, \mathbf{v}) \in Q \times \mathbb{N}^d$

Vector addition systems with states (VASS)

(d, Q, T), where $T \subseteq Q \times \mathbb{Z}^d \times Q$

Example: $d = 3$, $Q = \{p, q\}$



Configurations $q(\mathbf{v}) = (q, \mathbf{v}) \in Q \times \mathbb{N}^d$

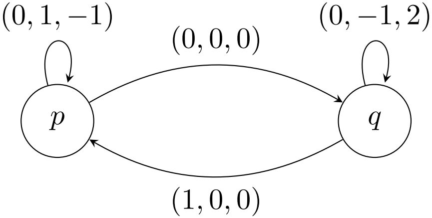
Example run:

$p(0, 0, 1) \rightarrow p(0, 1, 0) \rightarrow q(0, 1, 0) \rightarrow q(0, 0, 2) \rightarrow p(1, 0, 2)$

Vector addition systems with states (VASS)

(d, Q, T), where $T \subseteq Q \times \mathbb{Z}^d \times Q$

Example: $d = 3$, $Q = \{p, q\}$



Configurations $q(\mathbf{v}) = (q, \mathbf{v}) \in Q \times \mathbb{N}^d$

Example run:

$p(0,0,1) \to p(0,1,0) \to q(0,1,0) \to q(0,0,2) \to p(1,0,2)$

Notation: $p(0,0,1) \xrightarrow{*} p(1,0,2)$

Decision problems

Reachability problem:

GIVEN: VASS (d, Q, T) and configurations $p(\mathbf{u}), q(\mathbf{v})$

DECIDE: whether $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

Decision problems

Reachability problem:

$\textsc{Given}$: VASS (d, Q, T) and configurations $p(\mathbf{u}), q(\mathbf{v})$

$\textsc{Decide}$: whether $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

State reachability problem:

$\textsc{Given}$: VASS (d, Q, T) a configuration $p(\mathbf{u})$ and a control-state q

$\textsc{Decide}$: whether exists $\mathbf{v}$ s.t. $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

Decision problems

<u>Reachability problem</u>:

GIVEN: VASS (d, Q, T) and configurations $p(\mathbf{u}), q(\mathbf{v})$

DECIDE: whether $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

<u>State reachability problem</u>:

GIVEN: VASS (d, Q, T) a configuration $p(\mathbf{u})$ and a control-state q

DECIDE: whether exists $\mathbf{v}$ s.t. $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

- State reachability can be reduced to reachability

Decision problems

Reachability problem:

GIVEN: VASS (d, Q, T) and configurations $p(\mathbf{u}), q(\mathbf{v})$

DECIDE: whether $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

State reachability problem:

GIVEN: VASS (d, Q, T) a configuration $p(\mathbf{u})$ and a control-state q

DECIDE: whether exists $\mathbf{v}$ s.t. $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$?

- State reachability can be reduced to reachability
- Many other problems reduce to reachability or state reachability

VASS rechability problem

- Central problem in TCS:
  - formal languages,
  - logic,
  - concurrent systems,
  - process calculi,...
- Model of concurrency (Petri Nets) with extensive applications in modelling and analysis of:
  - hardware and software,
  - database systems,
  - chemical, biological and business processes.

Reachability state of art

Reachability state of art

1976 — EXPSPACE-hard (Lipton)

Reachability state of art

1976	EXPSPACE-hard (Lipton)
1981	Decidable (Mayr)

Reachability state of art

1976	EXPSPACE-hard (Lipton)
1981	Decidable (Mayr)
1982	Decidable (Kosaraju)

Reachability state of art

1976	EXPSPACE-hard (Lipton)
1981	Decidable (Mayr)
1982	Decidable (Kosaraju)
1992	Decidable (Lambert)

Reachability state of art

1976 — EXPSPACE-hard (Lipton)

1981 — Decidable (Mayr)

1982 — Decidable (Kosaraju)

1992 — Decidable (Lambert)

2009 — Decidable (Leroux)

2011 — Decidable (Leroux)

2012 — Decidable (Leroux)

Reachability state of art

1976 — EXPSPACE-hard (Lipton)

1981 — Decidable (Mayr)

1982 — Decidable (Kosaraju)

1992 — Decidable (Lambert)

2009 — Decidable (Leroux)

2011 — Decidable (Leroux)

2012 — Decidable (Leroux)

2015 — In $\mathbf{F}_{\omega^3}$ (Leroux and Schmitz)

Reachability state of art

1976	EXPSPACE-hard (Lipton)
1981	Decidable (Mayr)
1982	Decidable (Kosaraju)
1992	Decidable (Lambert)
2009	Decidable (Leroux)
2011	Decidable (Leroux)
2012	Decidable (Leroux)
2015	In $\mathbf{F}_{\omega^3}$ (Leroux and Schmitz)
2019	Nonelementary (this talk)

Reachability state of art

1976	EXPSPACE-hard (Lipton)
1981	Decidable (Mayr)
1982	Decidable (Kosaraju)
1992	Decidable (Lambert)
2009	Decidable (Leroux)
2011	Decidable (Leroux)
2012	Decidable (Leroux)
2015	In $\mathbf{F}_{\omega^3}$ (Leroux and Schmitz)
2019	Nonelementary (this talk)
2019?	In $\mathbf{F}_{\omega}$ (Leroux and Schmitz)

Reachability state of art

State reachability
Rackoff 1978
EXPSPACE-complete

1976 — EXPSPACE-hard (Lipton)

1981 — Decidable (Mayr)

1982 — Decidable (Kosaraju)

1992 — Decidable (Lambert)

2009 — Decidable (Leroux)

2011 — Decidable (Leroux)

2012 — Decidable (Leroux)

2015 — In $\mathbf{F}_{\omega^3}$ (Leroux and Schmitz)

2019 — Nonelementary (this talk)

2019? — In $\mathbf{F}_{\omega}$ (Leroux and Schmitz)

Counter programs

Counter programs

- Operations over bounded counters $\bar{\mathsf{x}}$:
  $\bar{\mathsf{x}} \mathrel{+}= 1$
  $\bar{\mathsf{x}} \mathrel{-}= 1$
  zero? $\bar{\mathsf{x}}$
  max? $\bar{\mathsf{x}}$
- Operations over an unbounded counter x:
  $\mathsf{x} \mathrel{+}= 1$
  $\mathsf{x} \mathrel{-}= 1$
- Non deterministic **loop**
- A last operation **halt if** $\mathsf{x}_1, \ldots, \mathsf{x}_n = 0$

Semantics

A B-run is a run such that:

- bounded counters ranges in $\{0, \ldots, B\}$,
- unbounded counters ranges in $\mathbb{N}$.

A run is **complete** if it:

- starts with zero in every counter, and
- ends by executing the last **halt if** $\mathsf{x}_1, \ldots, \mathsf{x}_n = 0$.

Counter Programs = VASS

Counter Programs = VASS

Reachability problem (for counter programs):

GIVEN: A counter program and a bound B.

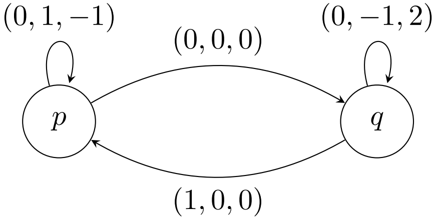
DECIDE: Does it have a complete B-run ?

Counter Programs = VASS

Reachability problem (for counter programs):

GIVEN: A counter program and a bound B.

DECIDE: Does it have a complete B-run ?

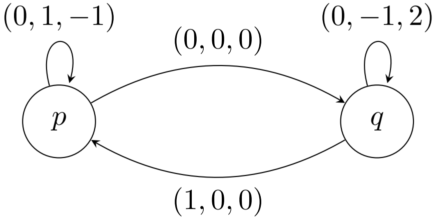


Counter Programs = VASS

Reachability problem (for counter programs):

Given: A counter program and a bound B.

Decide: Does it have a complete B-run ?



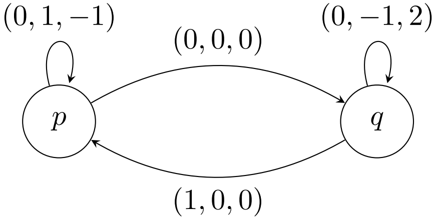
$p(0,0,1) \xrightarrow{*} p(1,0,2)$?

Counter Programs = VASS

<u>Reachability problem (for counter programs)</u>:

GIVEN: A counter program and a bound B.

DECIDE: Does it have a complete B-run ?



$p(0,0,1) \xrightarrow{*} p(1,0,2)$?

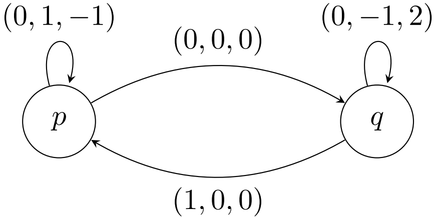
z += 1
loop
    loop
        y += 1   z −= 1
    loop
        y −= 1   z += 2
    x += 1
x −= 1   z −= 2
halt if x, y, z = 0.

Counter Programs = VASS

Reachability problem (for counter programs):

GIVEN: A counter program and a bound B.

DECIDE: Does it have a complete B-run ?



$p(0,0,1) \xrightarrow{*} p(1,0,2)$?

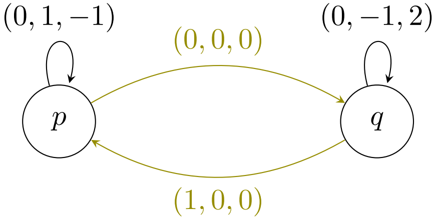
```
z += 1
loop
    loop
        y += 1    z −= 1
    loop
        y −= 1    z += 2
    x += 1
x −= 1    z −= 2
halt if x, y, z = 0.
```

Counter Programs = VASS

Reachability problem (for counter programs):

GIVEN: A counter program and a bound B.

DECIDE: Does it have a complete B-run ?



$p(0,0,1) \xrightarrow{*} p(1,0,2)$?

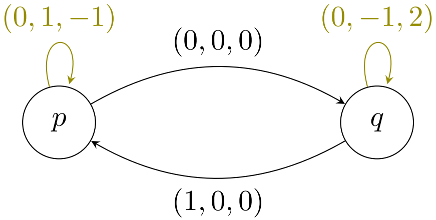
z $+=$ 1
loop
    loop
        y $+=$ 1   z $-=$ 1
    loop
        y $-=$ 1   z $+=$ 2
    x $+=$ 1
x $-=$ 1   z $-=$ 2
halt if x, y, z $= 0$.

Counter Programs = VASS

Reachability problem (for counter programs):

GIVEN: A counter program and a bound B.

DECIDE: Does it have a complete B-run ?



$p(0,0,1) \xrightarrow{*} p(1,0,2)$?

```
z += 1
loop
    loop
        y += 1    z -= 1
    loop
        y -= 1    z += 2
    x += 1
x -= 1    z -= 2
halt if x, y, z = 0.
```

Outline

- High level idea of the proof
- The factorial amplifier
- Composition operator

A TOWER-complete problem

The following problem is TOWER-complete (see Schmitz 2016)

GIVEN: A counter program without unbounded counters and n.

DECIDE: Does it have a complete $3 \underbrace{! \cdots !}_{n \text{ times}}$-run ?

B-computed relations

The relation B-computed in some counters $\mathsf{x}_1, \ldots, \mathsf{x}_l$ is the set of tuples of values after a complete B-run in those counters.

B-computed relations

The relation B-computed in some counters $\mathsf{x}_1, \ldots, \mathsf{x}_l$ is the set of tuples of values after a complete B-run in those counters.

loop
    $\bar{\mathsf{i}}$ += 1
    // assert $\bar{\mathsf{a}} = 0$
    loop
        x += 1 $\bar{\mathsf{a}}$ += 1
    max? $\bar{\mathsf{a}}$
    loop
        $\bar{\mathsf{a}}$ −= 1
    zero? $\bar{\mathsf{a}}$
max? $\bar{\mathsf{i}}$
halt

B-computed relations

The relation B-computed in some counters $\mathsf{x}_1, \ldots, \mathsf{x}_l$ is the set of tuples of values after a complete B-run in those counters.

$$
\left.
\begin{array}{l}
\textbf{loop} \\
\quad \bar{\mathsf{i}} \mathrel{+}= 1 \\
\quad \color{gray}{//\ \text{assert } \bar{\mathsf{a}} = 0} \\
\quad \textbf{loop} \\
\quad\quad \mathsf{x} \mathrel{+}= 1 \quad \bar{\mathsf{a}} \mathrel{+}= 1 \\
\quad \textbf{max? } \bar{\mathsf{a}} \\
\quad \textbf{loop} \\
\quad\quad \bar{\mathsf{a}} \mathrel{-}= 1 \\
\quad \textbf{zero? } \bar{\mathsf{a}}
\end{array}
\right\} \mathsf{x}{+} = B
$$

$$
\begin{array}{l}
\textbf{max? } \bar{\mathsf{i}} \\
\textbf{halt}
\end{array}
$$

B-computed relations

The relation B-computed in some counters $\mathsf{x}_1, \ldots, \mathsf{x}_l$ is the set of tuples of values after a complete B-run in those counters.

```
loop
    ī += 1
    // assert ā = 0
    loop
        x += 1    ā += 1
    max? ā                          x+ = B
    loop
        ā −= 1
    zero? ā
max? ī
halt
```

The relation B-computed in x is $\mathsf{x} = B^2$.

(B, R)-amplifier

A (B, R)-amplifier is a counter program that B-computes in $\mathsf{b}, \mathsf{c}, \mathsf{d}$ the relation $\mathsf{b} = R \ \wedge \ \mathsf{c} > 0 \ \wedge \ \mathsf{d} = \mathsf{c} \cdot R$.

(B, R)-amplifier

A (B, R)-amplifier is a counter program that B-computes in $\mathsf{b}, \mathsf{c}, \mathsf{d}$ the relation $\mathsf{b} = R \ \wedge\ \mathsf{c} > 0 \ \wedge\ \mathsf{d} = \mathsf{c} \cdot R$.

Example: Counter program $\mathcal{A}_3$

$\mathsf{b} \mathrel{+}= 3 \quad \mathsf{c} \mathrel{+}= 1 \quad \mathsf{d} \mathrel{+}= 3$
loop
  $\mathsf{c} \mathrel{+}= 1 \quad \mathsf{d} \mathrel{+}= 3$
halt

$\mathcal{A}_3$ is a $(0, 3)$-amplifier.

Simulation with amplifiers

Simulation with amplifiers

We provide a composition operator $\mathcal{A} \triangleright \mathcal{P}$ such that if:

- $\mathcal{A}$ is a (B, R)-amplifier.
- $\mathcal{P}$ is a counter program.

Then:

$$\text{Relations } R\text{-computed by } \mathcal{P} \\ = \\ \text{Relations } B\text{-computed by } \mathcal{A} \triangleright \mathcal{P}$$

Factorial Amplifier

There exists a counter program $\mathcal{F}$ that is a $(B, B!)$-amplifier for every $B > 0$ called the **factorial amplifier**.

Factorial Amplifier

There exists a counter program $\mathcal{F}$ that is a $(B, B!)$-amplifier for every $B > 0$ called the **factorial amplifier**.

$$\text{Relations } 3\overbrace{!\cdots!}^{n \text{ times}}\text{-computed by } \mathcal{P}$$
$$=$$
$$\text{Relations } 0\text{-computed by } \mathcal{A}_3 \triangleright \underbrace{\mathcal{F} \triangleright \cdots \triangleright \mathcal{F}}_{n \text{ times}} \triangleright \mathcal{P}$$

The Factorial Amplifier

to B-compute the relation $\mathsf{b} = B! \ \wedge \ \mathsf{c} > 0 \ \wedge \ \mathsf{d} = \mathsf{c} \cdot B!$

Main idea

Implement with a counter program:

$$n \cdot \prod_{1 \leq i < B} \frac{i+1}{i} = n \cdot B$$

A weak multiplier by $\frac{3}{2}$

$//$ assert $\mathsf{x} = x \quad \mathsf{x}' = x'$

loop

  $\mathsf{x} \mathrel{-}= 2 \quad \mathsf{x}' \mathrel{+}= 3$

loop

  $\mathsf{x}' \mathrel{-}= 1 \quad \mathsf{x} \mathrel{+}= 1$

$//$ assert $\mathsf{x} + \mathsf{x}' \leq \frac{3}{2}(x + x')$

$//$ assert $\mathsf{x} + \mathsf{x}' = \frac{3}{2}(x + x') \Rightarrow x' = 0$

A weak multiplier by $\frac{3}{2}$

$//$ assert $\mathsf{x} = x \quad \mathsf{x}' = x'$

loop

$\quad \mathsf{x} \mathrel{-}= 2 \quad \mathsf{x}' \mathrel{+}= 3$

loop

$\quad \mathsf{x}' \mathrel{-}= 1 \quad \mathsf{x} \mathrel{+}= 1$

$//$ assert $\mathsf{x} + \mathsf{x}' \leq \frac{3}{2}(x + x')$

$//$ assert $\mathsf{x} + \mathsf{x}' = \frac{3}{2}(x + x') \Rightarrow x' = 0$

x	x′	x + x′
15	0	15
13	3	16
11	6	17
9	9	18
7	12	19
5	15	20
3	18	21
1	21	$22 = \frac{3}{2}15 - \frac{1}{2}$
2	20	22
3	19	22
⋮	⋮	⋮
21	1	22
22	0	22

Implementing $n \cdot \prod_{1 \leq i < B} \frac{i+1}{i} = n \cdot B$

Implementing $n \cdot \prod_{1 \le i < B} \frac{i+1}{i} = n \cdot B$

$\bar{\mathsf{i}} \mathrel{+}= 1 \quad \mathsf{x} \mathrel{+}= 1 \quad \mathsf{y} \mathrel{+}= 1$

loop

$\quad \mathsf{x} \mathrel{+}= 1 \quad \mathsf{y} \mathrel{+}= 1$

loop

$\quad$ // assert $\mathsf{x} + \mathsf{x}' \le \mathsf{y} \cdot \bar{\mathsf{i}}$

$$
\left.
\begin{array}{l}
\textbf{loop} \\
\quad \mathsf{x} \mathrel{-}= \bar{\mathsf{i}} \quad \mathsf{x}' \mathrel{+}= \bar{\mathsf{i}} + 1 \\
\textbf{loop} \\
\quad \mathsf{x}' \mathrel{-}= 1 \quad \mathsf{x} \mathrel{+}= 1
\end{array}
\right\} \text{weak multiplier by } \frac{\bar{\mathsf{i}}+1}{\bar{\mathsf{i}}}
$$

$\quad$ // assert $\mathsf{x} + \mathsf{x}' \le \mathsf{y} \cdot (\bar{\mathsf{i}} + 1)$

$\quad \bar{\mathsf{i}} \mathrel{+}= 1$

max? $\bar{\mathsf{i}}$

loop

$\quad \mathsf{x} \mathrel{-}= \bar{\mathsf{i}} \quad \mathsf{y} \mathrel{-}= 1$

halt if $\mathsf{y} = 0$

How to simulate $\mathsf{x} \mathrel{-}= \bar{\mathsf{i}}$?

x is an unbounded counter
$\bar{\mathsf{i}}$ is a bounded counter and
$\bar{\mathsf{a}}$ is an auxiliary bounded counter assumed to be zero.

How to simulate $\mathsf{x} -= \bar{\mathsf{i}}$?

x is an unbounded counter
$\bar{\mathsf{i}}$ is a bounded counter and
$\bar{\mathsf{a}}$ is an auxiliary bounded counter assumed to be zero.

$//$ assert $\mathsf{x} = x \quad \bar{\mathsf{i}} = i \quad \bar{\mathsf{a}} = 0$
loop
$\quad \mathsf{x} -= 1 \quad \bar{\mathsf{i}} -= 1 \quad \bar{\mathsf{a}} += 1$
zero? $\bar{\mathsf{i}}$
$//$ assert $\mathsf{x} = x - i \quad \bar{\mathsf{i}} = 0 \quad \bar{\mathsf{a}} = i$
loop
$\quad \bar{\mathsf{i}} += 1 \quad \bar{\mathsf{a}} -= 1$
zero? $\bar{\mathsf{a}}$
$//$ assert $\mathsf{x} = x - i \quad \bar{\mathsf{i}} = i \quad \bar{\mathsf{a}} = 0$

How to simulate $\mathsf{x} \mathrel{+}= \bar{\mathsf{i}} + 1$?

x is an unbounded counter
$\bar{\mathsf{i}}$ is a bounded counter and
$\bar{\mathsf{a}}$ is an auxiliary bounded counter assumed to be zero.

How to simulate $\mathsf{x} \mathrel{+}= \bar{\mathsf{i}} + 1$?

x is an unbounded counter
$\bar{\mathsf{i}}$ is a bounded counter and
$\bar{\mathsf{a}}$ is an auxiliary bounded counter assumed to be zero.

$\mathsf{x} \mathrel{+}= 1$
loop
  $\mathsf{x} \mathrel{+}= 1 \quad \bar{\mathsf{i}} \mathrel{-}= 1 \quad \bar{\mathsf{a}} \mathrel{+}= 1$
zero? $\bar{\mathsf{i}}$
loop
  $\bar{\mathsf{i}} \mathrel{+}= 1 \quad \bar{\mathsf{a}} \mathrel{-}= 1$
zero? $\bar{\mathsf{a}}$

The factorial amplifier

The factorial amplifier

$\bar{\mathsf{i}} \mathrel{+}= 1 \quad \mathsf{b} \mathrel{+}= 1 \quad \mathsf{c} \mathrel{+}= 1 \quad \mathsf{d} \mathrel{+}= 1 \quad \mathsf{x} \mathrel{+}= 1 \quad \mathsf{y} \mathrel{+}= 1$

loop

$\quad \mathsf{c} \mathrel{+}= 1 \quad \mathsf{d} \mathrel{+}= 1 \quad \mathsf{x} \mathrel{+}= 1 \quad \mathsf{y} \mathrel{+}= 1$

loop

Multiply	x	d	c	b
by	$\frac{\bar{\mathsf{i}}+1}{\bar{\mathsf{i}}}$	$\frac{\bar{\mathsf{i}}+1}{\bar{\bar{\mathsf{i}}}}$	$\frac{1}{\bar{\bar{\mathsf{i}}}}$	$\bar{\mathsf{i}}+1$

$\quad \bar{\mathsf{i}} \mathrel{+}= 1$

max? $\bar{\mathsf{i}}$

loop

$\quad \mathsf{x} \mathrel{-}= \bar{\mathsf{i}} \quad \mathsf{y} \mathrel{-}= 1$

halt if $\mathsf{y} = 0$

Multiply	x	d	c	b
by	$\frac{\bar{\mathsf{i}}+1}{\bar{\mathsf{i}}}$	$\frac{\bar{\mathsf{i}}+1}{\bar{\mathsf{i}}}$	$\frac{1}{\bar{\mathsf{i}}}$	$\bar{\mathsf{i}}+1$

Multiply	x	d	c	b
by	$\frac{\bar{\mathrm{i}}+1}{\bar{\mathrm{i}}}$	$\frac{\bar{\mathrm{i}}+1}{\bar{\mathrm{i}}}$	$\frac{1}{\bar{\mathrm{i}}}$	$\bar{\mathrm{i}}+1$

// assert $x = d \leq y \cdot \bar{\mathrm{i}} \quad \wedge \quad c \geq \frac{y}{(\bar{\mathrm{i}}-1)!} \quad \wedge \quad b \leq \bar{\mathrm{i}}!$

loop
    $c \mathrel{-}= \bar{\mathrm{i}} \quad c' \mathrel{+}= 1$
    loop at most b **times**
        $d \mathrel{-}= \bar{\mathrm{i}} \quad x \mathrel{-}= \bar{\mathrm{i}} \quad d' \mathrel{+}= \bar{\mathrm{i}}+1$

} weak multiplier of x, d by $\frac{\bar{\mathrm{i}}+1}{\bar{\mathrm{i}}}$

loop
    $b \mathrel{-}= 1 \quad b' \mathrel{+}= \bar{\mathrm{i}}+1$
loop
    $b' \mathrel{-}= 1 \quad b \mathrel{+}= 1$

} weak multiplier of b by $\bar{\mathrm{i}}+1$

loop
    $c' \mathrel{-}= 1 \quad c \mathrel{+}= 1$
    loop at most b **times**
        $d' \mathrel{-}= 1 \quad d \mathrel{+}= 1 \quad x \mathrel{+}= 1$

Controled loops

loop at most b **times** <*body*>

b is an unbounded counter

b′ is an auxiliary unbounded counter

Controled loops

loop at most b **times** <*body*>

b is an unbounded counter

b′ is an auxiliary unbounded counter

loop

    $\mathsf{b} \mathrel{-}= 1 \quad \mathsf{b}' \mathrel{+}= 1$

loop

    $\mathsf{b}' \mathrel{-}= 1 \quad \mathsf{b} \mathrel{+}= 1$

    <*body*>

The composition operator

Composing with amplifiers

Composing with amplifiers

$$\text{Relations } R\text{-computed by } \mathcal{P}$$
$$=$$
$$\text{Relations } B\text{-computed by } \underbrace{\mathcal{A}}_{(B,R)\text{-amplifier}} \triangleright \mathcal{P}$$

Composing with amplifiers

Relations R-computed by $\mathcal{P}$

$$=$$

Relations B-computed by $\underbrace{\mathcal{A}}_{(B,R)\text{-amplifier}} \triangleright \mathcal{P}$

$$\boxed{\begin{array}{l}\text{body of } \mathcal{A} \\ \textbf{halt if } \mathsf{y}_1, \ldots = 0\end{array}} \triangleright \boxed{\begin{array}{l}\text{body of } \mathcal{P} \\ \textbf{halt if } \mathsf{z}_1, \ldots = 0\end{array}}$$

$$=$$

$$\boxed{\begin{array}{l}\text{body of } \mathcal{A} \\ \text{initialization body} \\ \text{modified body of } \mathcal{P} \\ \textbf{halt if } d, \mathsf{y}_1, \ldots, \mathsf{z}_1, \ldots = 0\end{array}}$$

Initialization body

$$\frac{\text{Invariants}}{b = R \wedge d = c \cdot R}$$

Encoding: We replace every bounded counter $\bar{\mathsf{x}}_i$ of $\mathcal{P}$ by two fresh unbounded counters x_i and x'_i satisfying $\mathsf{x}_i + \mathsf{x}'_i = R$.

Initialization body

Invariants
$b = R \land d = c \cdot R$

Encoding: We replace every bounded counter $\bar{\mathsf{x}}_i$ of $\mathcal{P}$ by two fresh unbounded counters x_i and x'_i satisfying $\mathsf{x}_i + \mathsf{x}'_i = R$.

$$\left.\begin{array}{l}
\textbf{loop} \\
\quad \mathsf{x}'_1 \mathrel{+}= 1 \quad \cdots \quad \mathsf{x}'_l \mathrel{+}= 1 \\
\quad \mathsf{b} \mathrel{-}= 1 \quad \mathsf{d} \mathrel{-}= 1 \\
\mathsf{c} \mathrel{-}= 1
\end{array}\right\} \begin{array}{l}
\text{c decreased by } 1 \text{ and} \\
\text{d by at most } R \\
\text{So if not complete d} > \text{c} \cdot R
\end{array}$$

Invariants OK	Invariants NOK
$d = c \cdot R \land \mathsf{x}_i + \mathsf{x}'_i = R$	$d > c \cdot R \land \mathsf{x}_i + \mathsf{x}'_i \leq R$

Modified body of $\mathcal{P}$

Replace $\bar{\mathsf{x}}_i \mathrel{+}= 1$ with $\mathsf{x}_i \mathrel{+}= 1 \quad \mathsf{x}'_i \mathrel{-}= 1$

Replace $\bar{\mathsf{x}}_i \mathrel{-}= 1$ with $\mathsf{x}_i \mathrel{-}= 1 \quad \mathsf{x}'_i \mathrel{+}= 1$

Invariants OK	Invariants NOK
$d = c \cdot R \wedge \mathsf{x}_i + \mathsf{x}'_i = R$	$d > c \cdot R \wedge \mathsf{x}_i + \mathsf{x}'_i \leq R$

Modified body of $\mathcal{P}$

Replace $\bar{\mathsf{x}}_i \mathrel{+}= 1$ with $\mathsf{x}_i \mathrel{+}= 1 \quad \mathsf{x}'_i \mathrel{-}= 1$

Replace $\bar{\mathsf{x}}_i \mathrel{-}= 1$ with $\mathsf{x}_i \mathrel{-}= 1 \quad \mathsf{x}'_i \mathrel{+}= 1$

Replace **zero?** $\bar{\mathsf{x}}_i$ with

loop

$\mathsf{x}_i \mathrel{+}= 1 \quad \mathsf{x}'_i \mathrel{-}= 1$

$\mathsf{d} \mathrel{-}= 1$

$\mathsf{c} \mathrel{-}= 1$

loop

$\mathsf{x}_i \mathrel{-}= 1 \quad \mathsf{x}'_i \mathrel{+}= 1$

$\mathsf{d} \mathrel{-}= 1$

$\mathsf{c} \mathrel{-}= 1$

Invariants OK	Invariants NOK
$d = c \cdot R \wedge \mathsf{x}_i + \mathsf{x}'_i = R$	$d > c \cdot R \wedge \mathsf{x}_i + \mathsf{x}'_i \leq R$

Modified body of $\mathcal{P}$

Replace $\bar{\mathsf{x}}_i \mathrel{+}= 1$ with $\mathsf{x}_i \mathrel{+}= 1 \quad \mathsf{x}'_i \mathrel{-}= 1$

Replace $\bar{\mathsf{x}}_i \mathrel{-}= 1$ with $\mathsf{x}_i \mathrel{-}= 1 \quad \mathsf{x}'_i \mathrel{+}= 1$

Replace **zero?** $\bar{\mathsf{x}}_i$ with

loop

  $\mathsf{x}_i \mathrel{+}= 1 \quad \mathsf{x}'_i \mathrel{-}= 1$

  $\mathsf{d} \mathrel{-}= 1$

$\mathsf{c} \mathrel{-}= 1$

loop

  $\mathsf{x}_i \mathrel{-}= 1 \quad \mathsf{x}'_i \mathrel{+}= 1$

  $\mathsf{d} \mathrel{-}= 1$

$\mathsf{c} \mathrel{-}= 1$

c decreased by 2 and d by at most $2R$

So if not complete $\mathsf{d} > \mathsf{c} \cdot R$

Invariants OK	Invariants NOK
$d = c \cdot R \wedge \mathsf{x}_i + \mathsf{x}'_i = R$	$d > c \cdot R \wedge \mathsf{x}_i + \mathsf{x}'_i \leq R$

To sum up

To sum up

We obtain that way a composition operator $\triangleright$ such that:

$$\text{Relations } 3\overbrace{!\cdots!}^{n \text{ times}}\text{-computed by } \mathcal{P}$$
$$=$$
$$\text{Relations } 0\text{-computed by } \mathcal{A}_3 \triangleright \underbrace{\mathcal{F} \triangleright \cdots \triangleright \mathcal{F}}_{n \text{ times}} \triangleright \mathcal{P}$$

Conclusion

Conclusion

- Reachability problem $>>>$ State reachability problem

Conclusion

- Reachability problem $>>>$ State reachability problem

- Plethora of problems are not elementary

In formal languages, logic, concurrent systems, process calculi,...

Conclusion

- Reachability problem $>>>$ State reachability problem

- Plethora of problems are not elementary

In formal languages, logic, concurrent systems, process calculi,...

- We can do h-EXPSPACE-hardness in dimension $h + 13$ (so fixed)

Conclusion

- Reachability problem $>>>$ State reachability problem

- Plethora of problems are not elementary

In formal languages, logic, concurrent systems, process calculi,...

- We can do h-EXPSPACE-hardness in dimension $h + 13$ (so fixed)

Can we do Tower in fixed dimension?

Conclusion

- Reachability problem $>>>$ State reachability problem

- Plethora of problems are not elementary

In formal languages, logic, concurrent systems, process calculi,...

- We can do h-EXPSPACE-hardness in dimension $h + 13$ (so fixed)

Can we do Tower in fixed dimension?

- The complexity is still open

Between Tower ($\mathbf{F}_3$) and Ackermann ($\mathbf{F}_\omega$)

Conclusion

- Reachability problem $>>>$ State reachability problem

- Plethora of problems are not elementary

In formal languages, logic, concurrent systems, process calculi,...

- We can do h-EXPSPACE-hardness in dimension $h + 13$ (so fixed)

Can we do Tower in fixed dimension?

- The complexity is still open

Between Tower ($\mathbf{F}_3$) and Ackermann ($\mathbf{F}_\omega$)

- Can we improve lower bounds of Pushdown-VASS, BVASS. . . ?

Conclusion

- Reachability problem $>>>$ State reachability problem

- Plethora of problems are not elementary

In formal languages, logic, concurrent systems, process calculi,...

- We can do h-EXPSPACE-hardness in dimension $h+13$ (so fixed)

Can we do Tower in fixed dimension?

- The complexity is still open

Between Tower ($\mathbf{F}_3$) and Ackermann ($\mathbf{F}_\omega$)

- Can we improve lower bounds of Pushdown-VASS, BVASS. . . ?

- This originated from studying 1-Pushdown-VASS