Reachabilty in Succinct and Parametric One-Counter Automata

C. Haase S. Kreutzer J. Ouaknine J. Worrell

Oxford University Computing Laboratory

ACTS
Feb, 2010

Parameters Everywhere

Boltzman's constant	k
Planck's constant	$\hbar$
Speed of light	c
Gravitational constant	G

...

Parameters Everywhere

Boltzman's constant	k
Planck's constant	$\hbar$
Speed of light	c
Gravitational constant	G

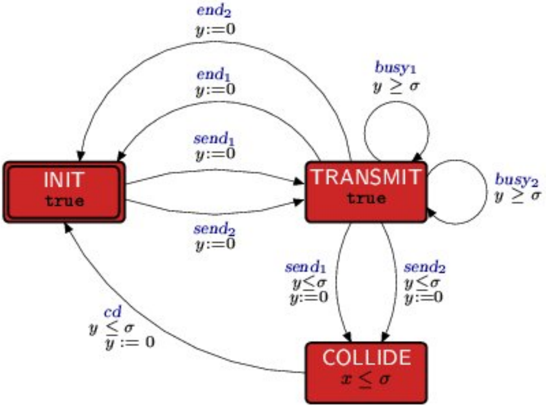
...

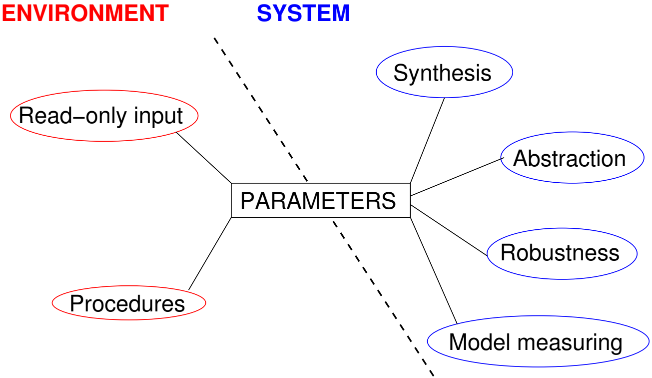
Parameters Everywhere

Boltzman's constant	k
Planck's constant	$\hbar$
Speed of light	c
Gravitational constant	G

...

A More Tractable Example

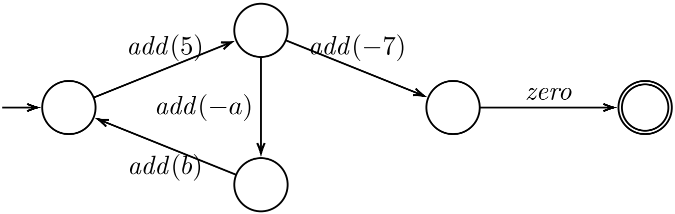



ENVIRONMENT
SYSTEM
Synthesis
Read–only input
Abstraction
PARAMETERS
Robustness
Procedures
Model measuring

Parametric State Machines

- Flat counter machines with parameters
  (Bozga, Iosif, Lakhnech 06)
- Reversal-bounded counter machines with read-only input
  (Dang, Ibarra 93 ; . . .)
- Timed automata with parametric guards
  (Alur, Henzinger, Vardi 93 ; André, Encrenaz, Fribourg 09)
- Counter machines with weights/costs
  (Xie, Dang, Ibarra 03)

Parametric One-Counter Automata

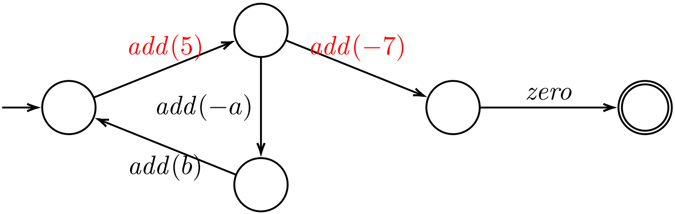


One-counter automata: NFA with one counter over $\mathbb{N}$

Succinct: Numbers encoded in binary

Parametric: Increment and decrement counter by parametric values

Parametric One-Counter Automata

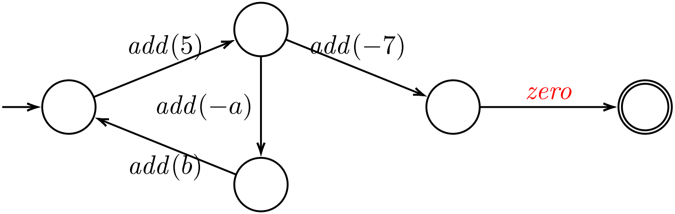


One-counter automata: NFA with one counter over $\mathbb{N}$

Succinct: Numbers encoded in binary

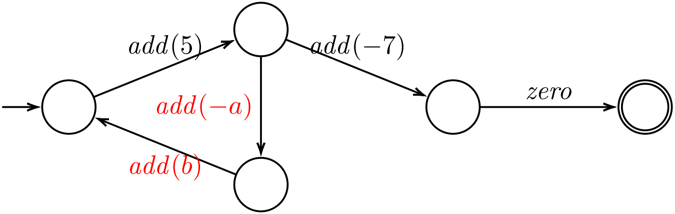
Parametric: Increment and decrement counter by parametric values

Parametric One-Counter Automata



One-counter automata:	NFA with one counter over $\mathbb{N}$
Succinct:	Numbers encoded in binary
Parametric:	Increment and decrement counter by parametric values

Parametric One-Counter Automata

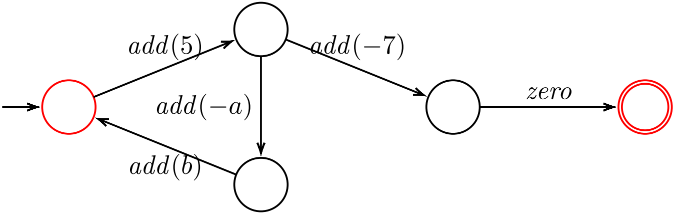


One-counter automata: NFA with one counter over $\mathbb{N}$

Succinct: Numbers encoded in binary

Parametric: Increment and decrement counter by parametric values

Parametric One-Counter Automata



Are there values for the parameters such that a final configuration is reachable from an initial configuration?

Main result

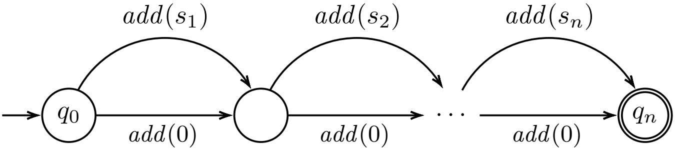
Theorem

The reachability problem for parametric one-counter automata is NP*-complete.*

NP-Hardness

Reduction from SUBSETSUM:

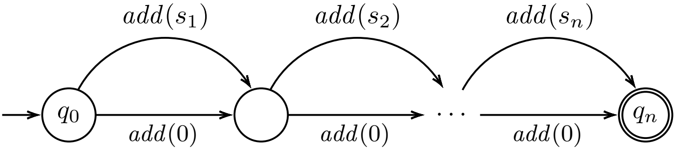
Instance: $S = \{s_1, s_2 \ldots, s_n\} \subseteq \mathbb{N}$ and target $t \in \mathbb{N}$
Question: Is there $S' \subseteq S$ such that $\sum_{s \in S'} s = t$?



NP-Hardness

Reduction from SUBSETSUM:

Instance: $S = \{s_1, s_2 \ldots, s_n\} \subseteq \mathbb{N}$ and target $t \in \mathbb{N}$
Question: Is there $S' \subseteq S$ such that $\sum_{s \in S'} s = t$?



Problem becomes NLOGSPACE-complete when numbers are encoded in unary (Lafourcade *et al.*, 2004)

Presburger Arithmetic

- First-order theory of the natural numbers with addition is decidable (Presburger 29)
- Adding multiplication or divisibility leads to undecidability of satisfiability (Gödel 31, Robinson 49)
- Existential fragment of PA with divisibility is decidable (Lipshitz 78)
  - Terms: linear polynomials $A(\vec{x}) = a_0 + a_1 x_1 + \ldots + a_n x_n$
  - Atomic formulas: $A(\vec{x}) \leq B(\vec{x})$ and $A(\vec{x}) | B(\vec{x})$
  - Formulas: $\exists x_1 \cdots \exists x_n : \varphi(\vec{x})$

Presburger+Divisibility –> Reachability

Idea. Given $\varphi(\vec{x})$, construct counter machine $\mathcal{C}_\varphi$ with parameters $\vec{x}$ such that $\varphi(\vec{x})$ iff $(q_s, 0) \rightsquigarrow (q_t, 0)$:

Presburger+Divisibility –> Reachability

Idea. Given $\varphi(\vec{x})$, construct counter machine $\mathcal{C}_\varphi$ with parameters $\vec{x}$ such that $\varphi(\vec{x})$ iff $(q_s, 0) \rightsquigarrow (q_t, 0)$:

- $\varphi_1 \wedge \varphi_2$: sequential composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$

Presburger+Divisibility –> Reachability

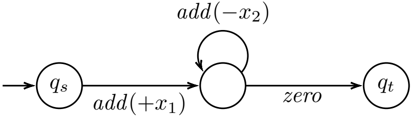
Idea. Given $\varphi(\vec{x})$, construct counter machine $\mathcal{C}_\varphi$ with parameters $\vec{x}$ such that $\varphi(\vec{x})$ iff $(q_s, 0) \rightsquigarrow (q_t, 0)$:

- $\varphi_1 \wedge \varphi_2$: sequential composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$
- $\varphi_1 \vee \varphi_2$: parallel composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$

Presburger+Divisibility –> Reachability

Idea. Given $\varphi(\vec{x})$, construct counter machine $\mathcal{C}_\varphi$ with parameters $\vec{x}$ such that $\varphi(\vec{x})$ iff $(q_s, 0) \rightsquigarrow (q_t, 0)$:

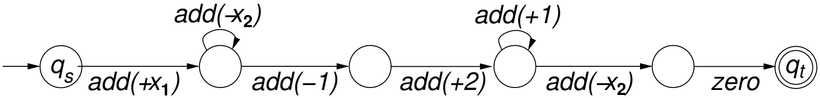
- $\varphi_1 \wedge \varphi_2$: sequential composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$
- $\varphi_1 \vee \varphi_2$: parallel composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$
- $x_1 \mid x_2$



Presburger + Divisibility –> Reachability

Idea. Given formula $\varphi(\vec{x})$, construct counter machine $\mathcal{C}_\varphi$ such that $\varphi(\vec{x})$ holds iff $(q_s, 0) \rightsquigarrow (q_t, 0)$ in $\mathcal{C}_\varphi$.

- $\varphi_1 \wedge \varphi_2$: sequential composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$
- $\varphi_1 \vee \varphi_2$: parallel composition of $\mathcal{C}_{\varphi_1}$ and $\mathcal{C}_{\varphi_2}$
- $x_2 \nmid x_1$



NP-Hardness Again

- **Theorem** (Manders, Adelman 76). The following problem is NP-complete:

  Given integers α, β, γ does there exist $x \leq \gamma$ such that

  $$x^2 \equiv \alpha \pmod{\beta}$$

NP-Hardness Again

- **Theorem** (Manders, Adelman 76). The following problem is NP-complete:

  Given integers α, β, γ does there exist $x \leq \gamma$ such that

  $$x^2 \equiv \alpha \pmod{\beta}$$

- Easily encoded into Presburger arithmetic with divisibility

NP-Hardness Again

- **Theorem** (Manders, Adelman 76). The following problem is NP-complete:

  Given integers α, β, γ does there exist $x \leq \gamma$ such that

  $$x^2 \equiv \alpha \pmod{\beta}$$

- Easily encoded into Presburger arithmetic with divisibility
- Reachability is NP-hard on counter machines even if we fix the underlying graph of states and transitions.

Words of Wisdom

Words of Wisdom

"If you can't solve a problem, there is an easier problem you can't solve." - ***George Pólya***

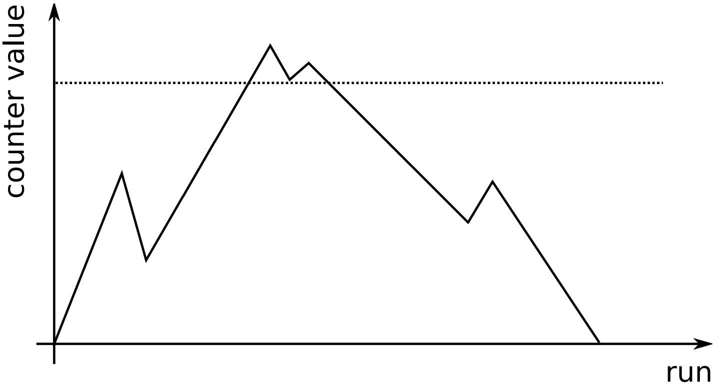
The non-parametric case

NP-Membership of Reachability

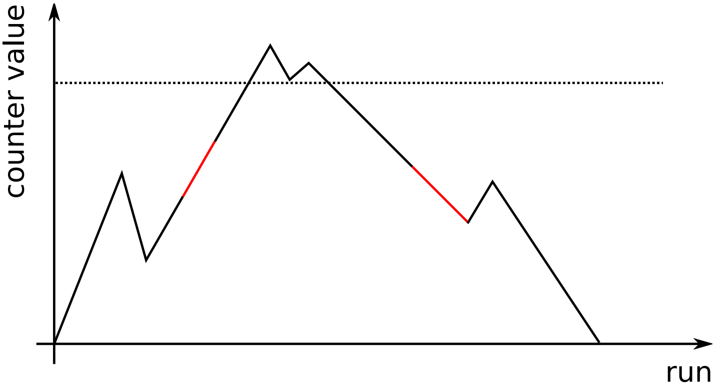
Three stages to show membership in NP:

1. Establish a bound on the length of a run
2. Find certificate of a run of polynomial size
3. Ensure certificate can be verified in non-deterministic polynomial time

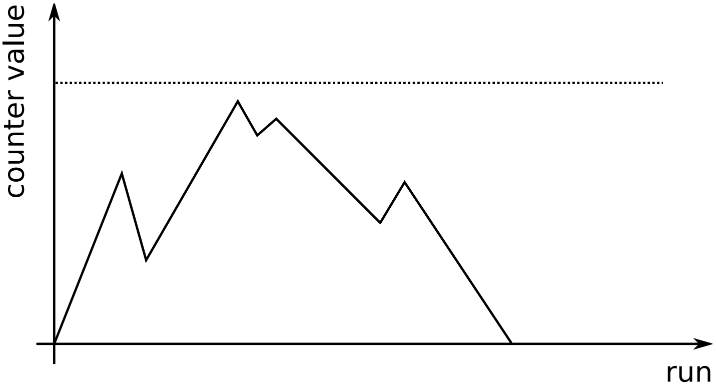
Truncating Runs (Lafourcade *et al.*, 2004)



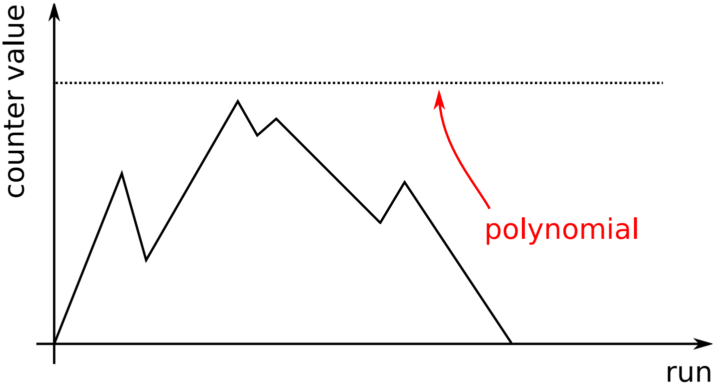
Truncating Runs (Lafourcade *et al.*, 2004)



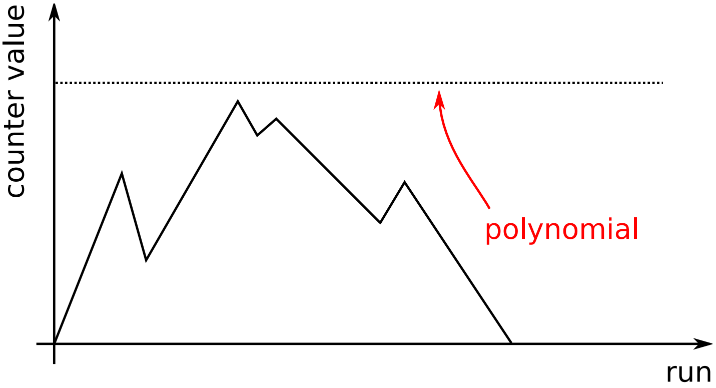
Truncating Runs (Lafourcade *et al.*, 2004)



Truncating Runs (Lafourcade *et al.*, 2004)



Truncating Runs (Lafourcade *et al.*, 2004)



$\rightsquigarrow$ PSPACE upper bound for reachability

NP-Membership of Reachability

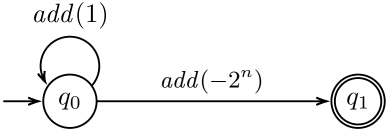
Three stages to show membership in NP:

1. Establish a bound on the length of a run

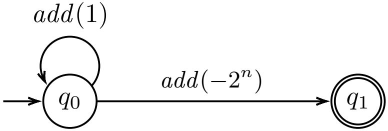
2. Find certificate of polynomial size of a run

3. Ensure certificate can be verified in non-deterministic polynomial time

Runs of Exponential Length

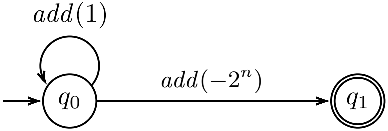


Runs of Exponential Length



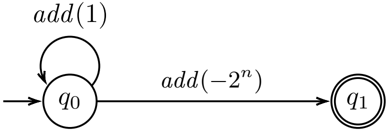
$(q_0, 0)$

Runs of Exponential Length



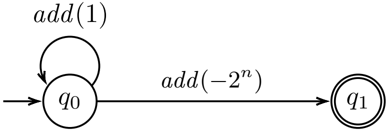
$(q_0, 0) \rightarrow (q_0, 1)$

Runs of Exponential Length



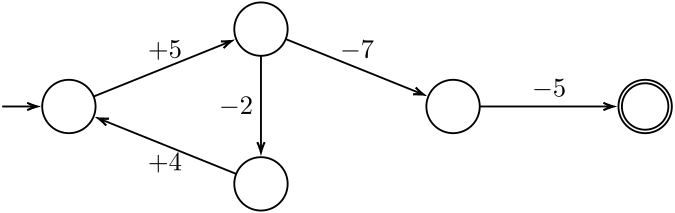
$$(q_0, 0) \rightarrow (q_0, 1) \rightarrow (q_0, 2)$$

Runs of Exponential Length

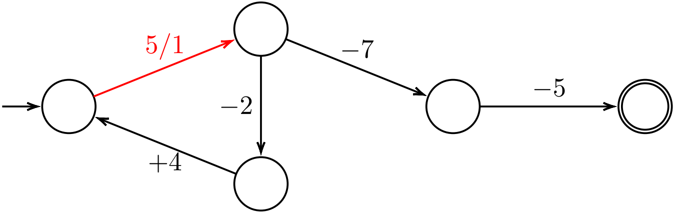


$$(q_0, 0) \rightarrow (q_0, 1) \rightarrow (q_0, 2) \rightarrow \cdots \rightarrow (q_1, 2^n) \rightarrow (q_1, 0)$$

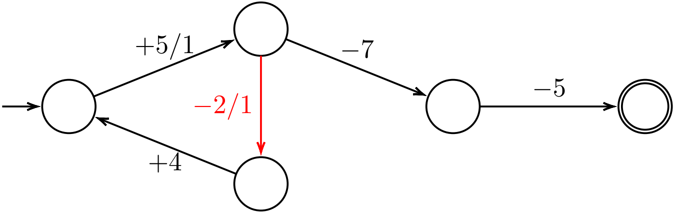
Flow Networks



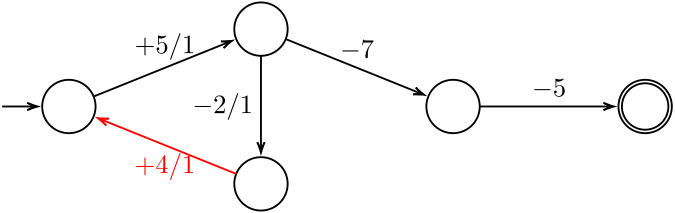
Flow Networks



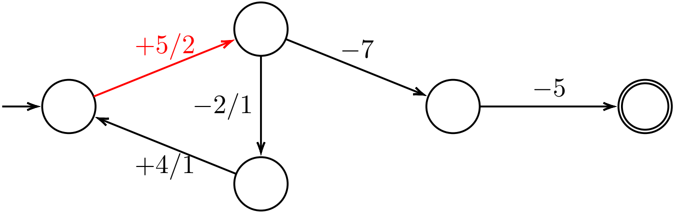
Flow Networks



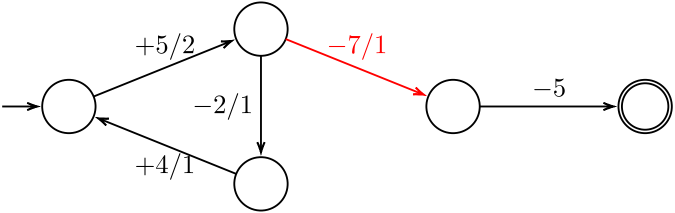
Flow Networks



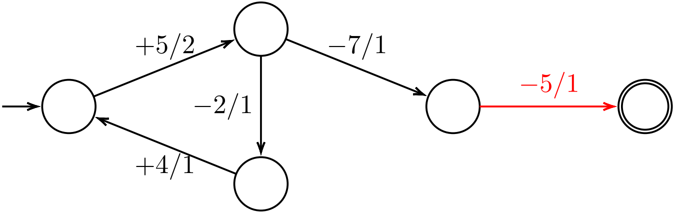
Flow Networks



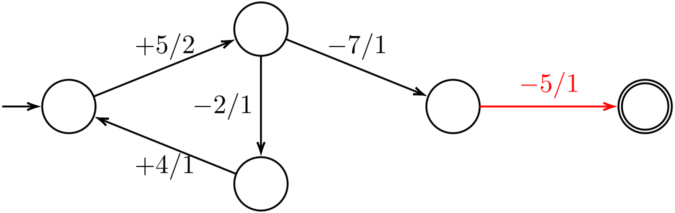
Flow Networks



Flow Networks

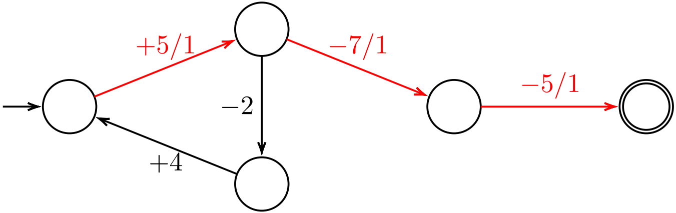


Flow Networks



⇝ assign to each edge the number of times it is taken:

Flow Networks



but flow network does not necessarily correspond to a run

NP-Membership of Reachability

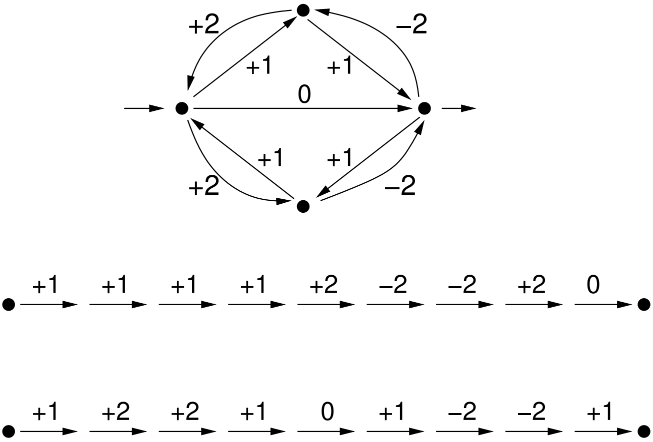
Three stages to show membership in NP:

1. Establish a bound on the length of a run
2. Find certificate of polynomial size of a run
3. Ensure certificate can be verified in non-deterministic polynomial time

Three Simple Cases

1. Flow network begins with a positive cycle and ends with a negative cycle
2. Flow network has no positive cycles
3. Flow network has no negative cycles

Positive Cycles and Positive Cycles



Positive Cycle and Negative Cycle

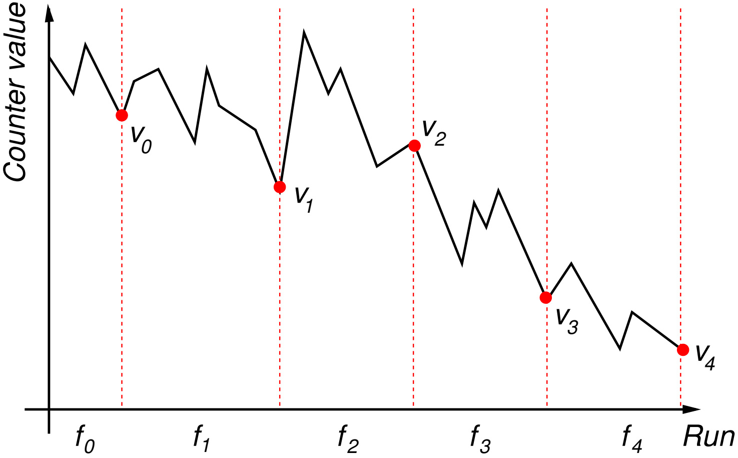
Three Simple Cases

1. Flow network begins with a positive cycle and ends with a negative cycle

2. Flow network has no positive cycles

3. Flow network has no negative cycles

No Positive Cycles



No Positive Cycles

- Guess **elimination order** on vertices
  - $v_0, v_1, \ldots, v_4$

No Positive Cycles

- Guess **elimination order** on vertices
  - $v_0, v_1, \ldots, v_4$
- Corresponding flow decomposition
  - $f = f_0 + f_1 + \cdots + f_4$

No Positive Cycles

- Guess **elimination order** on vertices
  - $v_0, v_1, \ldots, v_4$
- Corresponding flow decomposition
  - $f = f_0 + f_1 + \cdots + f_4$
- Counter never goes negative:
  - $\mathit{value}(f_0) \geq 0$
  - $\mathit{value}(f_0 + f_1) \geq 0$

  $\ldots$

Three Simple Cases

1. Flow network begins with a positive cycle and ends with a negative cycle

2. Flow network has no positive cycles

3. Flow network has no negative cycles

Decomposition Lemma

Lemma

If there is a path from the initial state to the final state, then there is a path that can be written as the sum of three flow networks $f^- + f^* + f^+$, *where*

- f^- *contains no positive cycle*
- f^+ *contains no negative cycle*
- f^* *has a positive cycle at the "beginning" and a negative cycle at the "end"*

Kirchhoff Certificates

Kirchhoff certificate guessed and verified in NP:

- Flows f^-, f^+ and f^* guessed in polynomial time
- Bellman-Ford algorithm checks in polynomial time non-existence of positive cycles in f^- and negative cycles in f^+
- Elimination orderings for f^+ and f^- guessed in polynomial time

$\leadsto$ NP-algorithm

NP-Membership of Reachability

Three stages to show membership in NP:

1. Establish a bound on the length of a run
2. Find certificate of polynomial size of a run
3. Ensure certificate can be verified in non-deterministic polynomial time

$\rightsquigarrow$ reachability for succinct one-counter automata is NP-complete

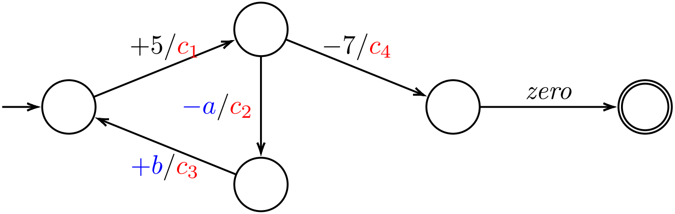
In Reality

In Reality

"It's only 10 pages in the LNCS style – we need another result!" - ***Christoph Haase***

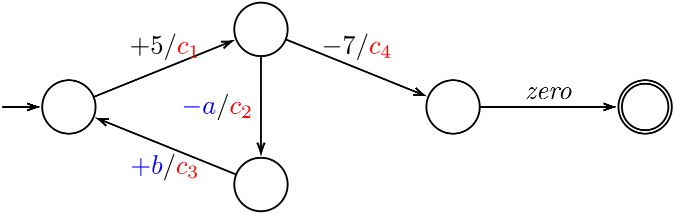
The parametric case

Symbolic Representation



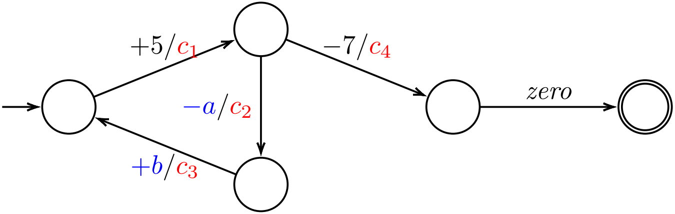
- Symbolic representation of Kirchhoff certificates

Symbolic Representation



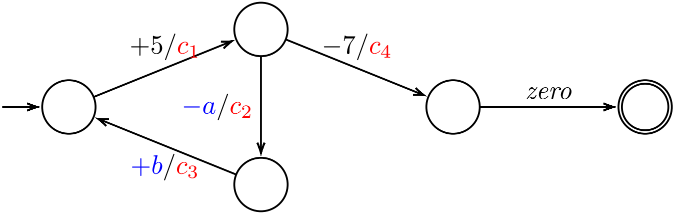
- Symbolic representation of Kirchhoff certificates
- Variables c_1, c_2, c_3, c_4 to represent flow

Symbolic Representation



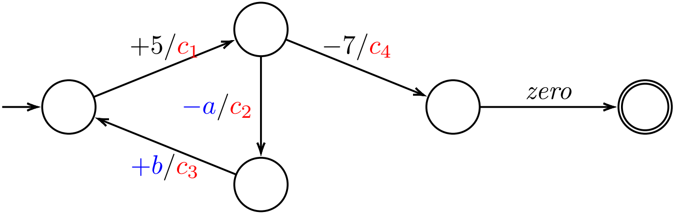
- Symbolic representation of Kirchhoff certificates
- Variables c_1, c_2, c_3, c_4 to represent flow
- Variables a, b to represent parameters

Symbolic Representation



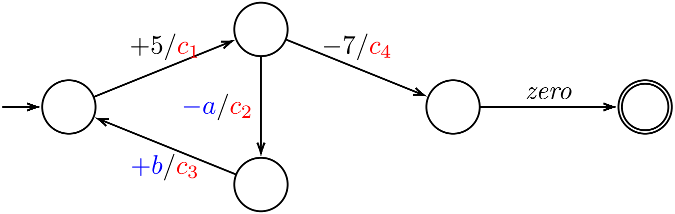
- Flow constraints: e.g. $c_1 = c_2 + c_4$

Symbolic Representation



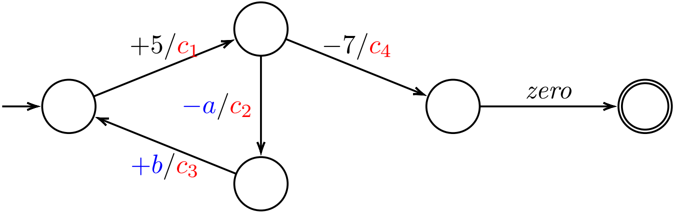
- Flow constraints: e.g. $c_1 = c_2 + c_4$
- Cycle constraints: e.g. $b - a + 5 > 0$

Symbolic Representation



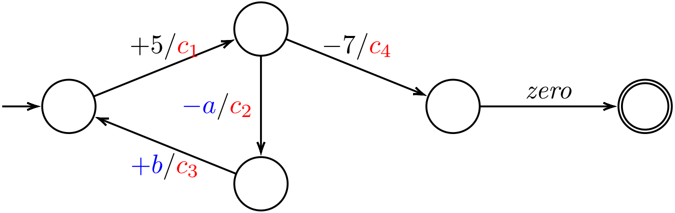
- Flow constraints: e.g. $c_1 = c_2 + c_4$
- Cycle constraints: e.g. $b - a + 5 > 0$
- Value constraints: $value(f) > 0$

Symbolic Representation



Value constraints:

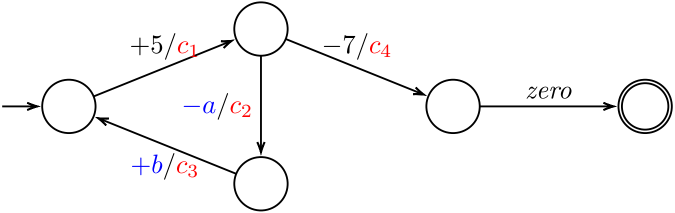
Symbolic Representation



Value constraints:

- $value(f) = 5 \cdot c_1 - a \cdot c_2 + b \cdot c_3 - 7 \cdot c_4$

Symbolic Representation



Value constraints:

- $\mathit{value}(f) = 5 \cdot c_1 - a \cdot c_2 + b \cdot c_3 - 7 \cdot c_4$
- Quadratic Diophantine equation

Flow Networks and Diophantine Equations

Some systems of quadratic Diophantine equations are decidable:

$$\begin{aligned} R_1 &= y_1 A_1(\vec{x}) + B_1(\vec{x}) \\ &\vdots \\ R_k &= y_k A_k(\vec{x}) + B_k(\vec{x}) \end{aligned}$$

Given $P \subseteq \mathbb{Z}^k$ Presburger definable, ask

$$\exists \vec{x} \exists \vec{y} P(R_1, \ldots, R_k)?$$

Flow Networks and Diophantine Equations

Some systems of quadratic Diophantine equations are decidable:

$$\begin{aligned} R_1 &= y_1 A_1(\vec{x}) + B_1(\vec{x}) \\ &\vdots \\ R_k &= y_k A_k(\vec{x}) + B_k(\vec{x}) \end{aligned}$$

Given $P \subseteq \mathbb{Z}^k$ Presburger definable, ask

$$\exists \vec{x} \exists \vec{y} P(R_1, \ldots, R_k)?$$

…translate to sentence in Presburger arithmetic with divisibility:

Summary

- Satisfiability in the existential fragment of Presburger arithmetic with divisibility is NP-complete (Lipshitz, 1976)

Summary

- Satisfiability in the existential fragment of Presburger arithmetic with divisibility is NP-complete (Lipshitz, 1976)
- All conditions of a reachability certificate can be encoded in a sentence of polynomial size in this logic

Summary

- Satisfiability in the existential fragment of Presburger arithmetic with divisibility is NP-complete (Lipshitz, 1976)
- All conditions of a reachability certificate can be encoded in a sentence of polynomial size in this logic
- Satisfiability in this fragment is inter-reducible with reachability in parametric one-counter automata

Summary

- Satisfiability in the existential fragment of Presburger arithmetic with divisibility is NP-complete (Lipshitz, 1976)
- All conditions of a reachability certificate can be encoded in a sentence of polynomial size in this logic
- Satisfiability in this fragment is inter-reducible with reachability in parametric one-counter automata

Theorem

The reachability problem for parametric one-counter automata is NP-complete.