The Complexity of Flat Freeze LTL

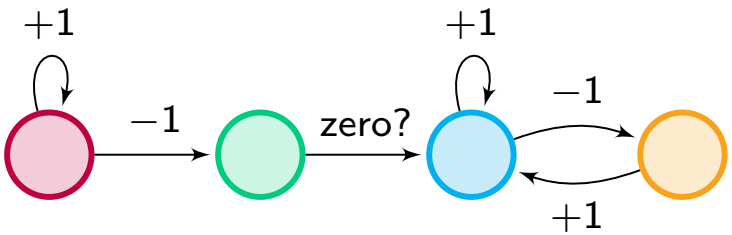
Benedikt Bollig [1], Karin Quaas [2], Arnaud Sangnier [3]

[1] Université Paris-Saclay, [2] Universität Leipzig, [3] Université Paris Diderot

CONCUR 2017

Model Checking One-Counter Machines & Freeze LTL

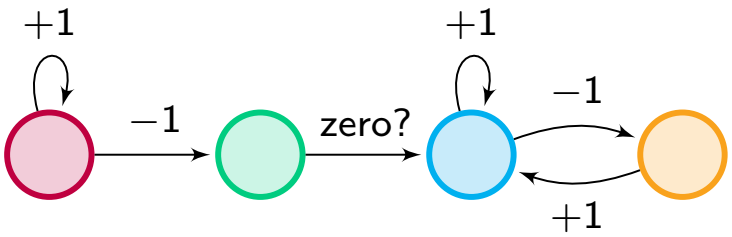
One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

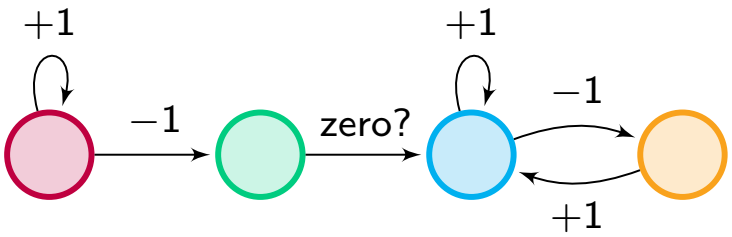
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X} p_2 \wedge {\downarrow} r.(\mathsf{GF}(= r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

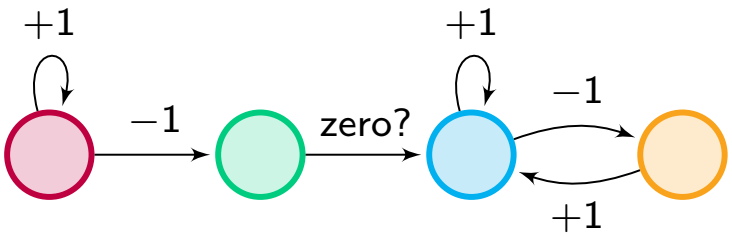
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X} p_2 \wedge {\downarrow} r.(\mathsf{GF}(= r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

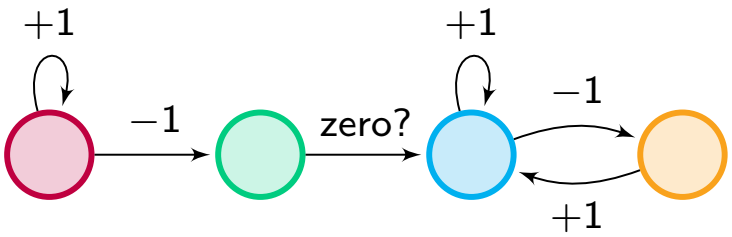
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X}p_2 \wedge {\downarrow}r.(\mathsf{GF}(=r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

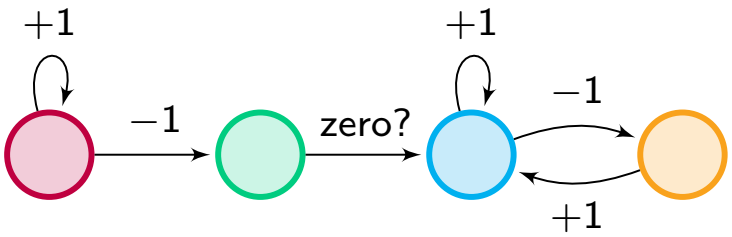
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X}p_2 \wedge {\downarrow}r.(\mathsf{GF}(=r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

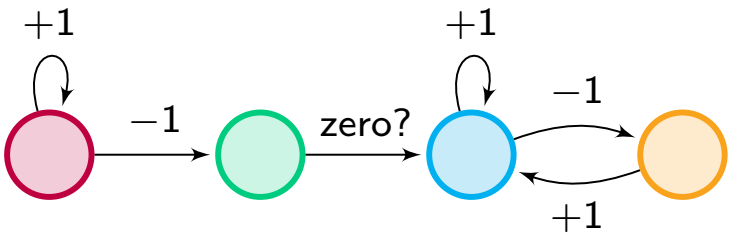
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X}p_2 \wedge {\downarrow}r.(\mathsf{GF}(=r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

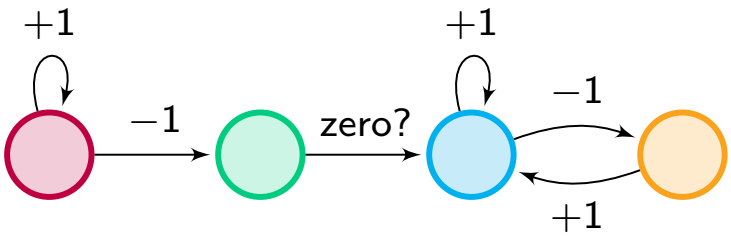
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X}p_2 \wedge \downarrow r.(\mathsf{GF}(=r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

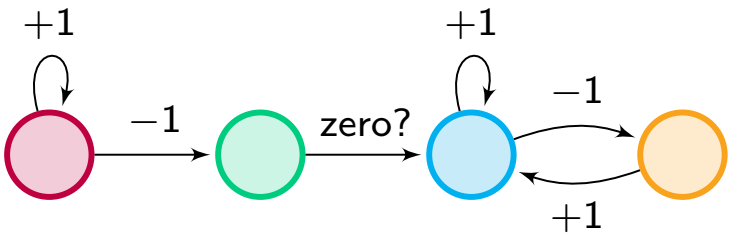
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X} p_2 \wedge {\downarrow} r.(\mathsf{GF}(= r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

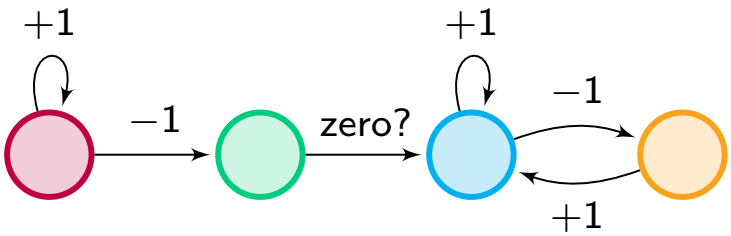
Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X} p_2 \wedge {\downarrow} r.(\mathsf{GF}(= r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X}p_2 \wedge \downarrow r.(\mathsf{GF}(=r \wedge p_3)))$$

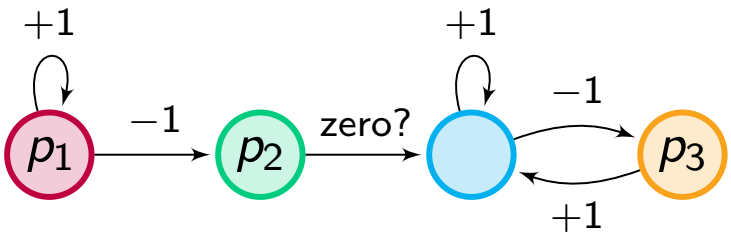
Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

(Existential) Model Checking

Given a one-counter machine $\mathcal{A}$ and a formula φ, does there exist a computation of $\mathcal{A}$ that satisfies φ?

Model Checking One-Counter Machines & Freeze LTL

One-Counter Machines



$0 \to 1 \to 0 \to 0 \to 1 \to 2(\to 1 \to 2)^{\omega}$

Freeze LTL

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{X}p_2 \wedge \downarrow r.(\mathsf{GF}(= r \wedge p_3)))$$

Finally, p_1 holds, and in the next position p_2 holds, and the current counter value (stored into r) will infinitely often reappear together with p_3.

(Existential) Model Checking

Given a one-counter machine $\mathcal{A}$ and a formula φ, does there exist a computation of $\mathcal{A}$ that satisfies φ?

Freeze LTL

The set of Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

Freeze LTL

The set of Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

Example

$\mathsf{F}\downarrow r.\mathsf{GF}{=}r$ Some counter value occurs infinitely often.

(where $\mathsf{F}\varphi \stackrel{\text{def}}{=} \mathsf{true}\mathsf{U}\varphi$, $\mathsf{G}\varphi \stackrel{\text{def}}{=} \neg\mathsf{F}\neg\varphi$)

Freeze LTL

The set of Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathrm{AP}$, $r \in \{r_1, r_2, \dots\}$

Example

$\mathsf{F}\downarrow r.\mathsf{GF}{=}r$ Some counter value occurs infinitely often.

$\mathsf{G}\downarrow r.\mathsf{GF}{=}r$ All counter values occur infinitely often.

(where $\mathsf{F}\varphi \stackrel{\text{def}}{=} \mathsf{true}\mathsf{U}\varphi$, $\mathsf{G}\varphi \stackrel{\text{def}}{=} \neg\mathsf{F}\neg\varphi$)

Freeze LTL

The set of Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

Example

$\mathsf{F}\downarrow r.\mathsf{GF}{=}r$ Some counter value occurs infinitely often.

$\mathsf{G}\downarrow r.\mathsf{GF}{=}r$ All counter values occur infinitely often.

(where $\mathsf{F}\varphi \stackrel{\text{def}}{=} \mathsf{true}\mathsf{U}\varphi$, $\mathsf{G}\varphi \stackrel{\text{def}}{=} \neg\mathsf{F}\neg\varphi$)

Theorem (Demri, Lazic, Sangnier 2010)

Model checking one-counter machines against formulas of Freeze LTL is undecidable, even if the formulas only use a single register.

Flat Freeze LTL

The set of Flat Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

Flat Freeze LTL

The set of Flat Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

$$\underbrace{\neg(\dots(\neg(}_{\mathsf{n}}\dots \varphi \mathsf{U} \psi \dots)) \Rightarrow \begin{cases} \mathsf{n} \text{ is even} \Rightarrow \downarrow r \text{ does not occur in } \varphi \\ \mathsf{n} \text{ is odd} \Rightarrow \downarrow r \text{ does not occur in } \psi \end{cases}$$

Flat Freeze LTL

The set of Flat Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

$$\underbrace{\neg(\dots(\neg(}_{\mathsf{n}}\dots \varphi \mathsf{U} \psi \dots)) \Rightarrow \begin{cases} \mathsf{n} \text{ is even} \Rightarrow \downarrow r \text{ does not occur in } \varphi \\ \mathsf{n} \text{ is odd} \Rightarrow \downarrow r \text{ does not occur in } \psi \end{cases}$$

(Intuition: bound the number of freeze operations $\downarrow r$)

Flat Freeze LTL

The set of Flat Freeze LTL formulas is defined by the grammar

$$\varphi ::= p \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \downarrow r.\varphi \mid {=}r$$

where $p \in \mathsf{AP}$, $r \in \{r_1, r_2, \dots\}$

$$\underbrace{\neg(\dots(\neg(}_{\mathsf{n}}\dots \varphi \mathsf{U} \psi \dots)) \Rightarrow \begin{cases} \mathsf{n} \text{ is even} \Rightarrow \downarrow r \text{ does not occur in } \varphi \\ \mathsf{n} \text{ is odd} \Rightarrow \downarrow r \text{ does not occur in } \psi \end{cases}$$

(Intuition: bound the number of freeze operations $\downarrow r$)

Example

$\mathsf{F}\downarrow r.\mathsf{GF}{=}r$ Some counter value occurs infinitely often. ✓

$\mathsf{G}\downarrow r.\mathsf{GF}{=}r$ All counter values occur infinitely often. ✗

(where $\mathsf{F}\varphi \stackrel{\text{def}}{=} \mathsf{true}\mathsf{U}\varphi$, $\mathsf{G}\varphi \stackrel{\text{def}}{=} \neg\mathsf{F}\neg\varphi$)

Flat Freeze LTL - History

- introduced by Demri and Sangnier, 2010

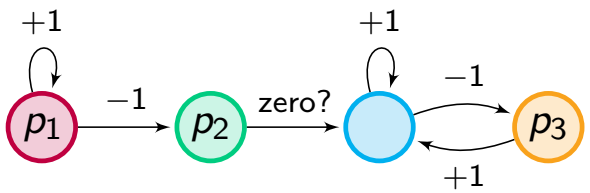
Flat Freeze LTL - History

- introduced by Demri and Sangnier, 2010
- main contribution:

Flat Freeze LTL - History

- introduced by Demri and Sangnier, 2010
- main contribution:

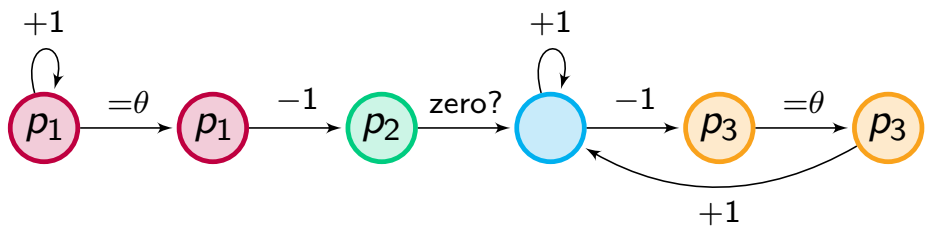
Model Checking



$\mathsf{F}(p_1 \wedge \mathsf{X}p_2) \wedge \downarrow r.(\mathsf{GF}(= r \wedge p_3))$

red. / exp.

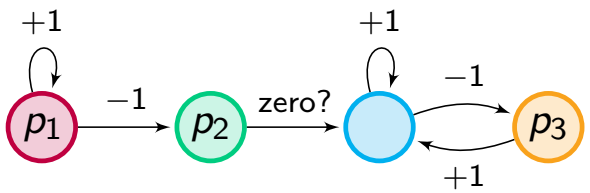
Reachability of
OCMs with parameterized tests



Flat Freeze LTL - History

- introduced by Demri and Sangnier, 2010
- main contribution:

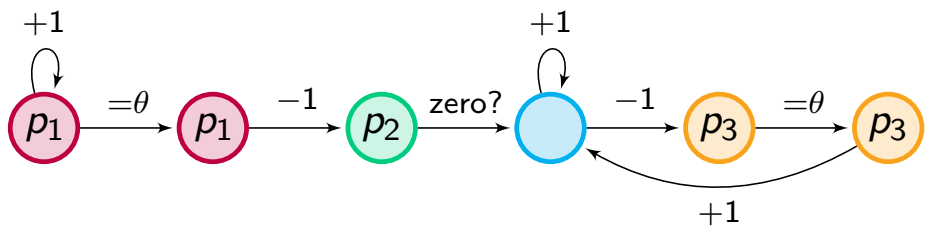
Model Checking



$\mathsf{F}(p_1 \wedge \mathsf{X}p_2) \wedge \downarrow r.(\mathsf{GF}(=r \wedge p_3))$

red.
exp.

Reachability of
OCMs with parameterized tests

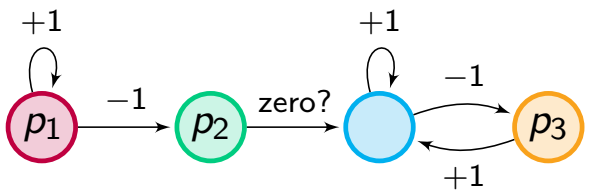


- decidability of model checking by Lechner, Mayr, Ouaknine, Pouly, Worrell, 2016

Flat Freeze LTL - History

- introduced by Demri and Sangnier, 2010
- main contribution:

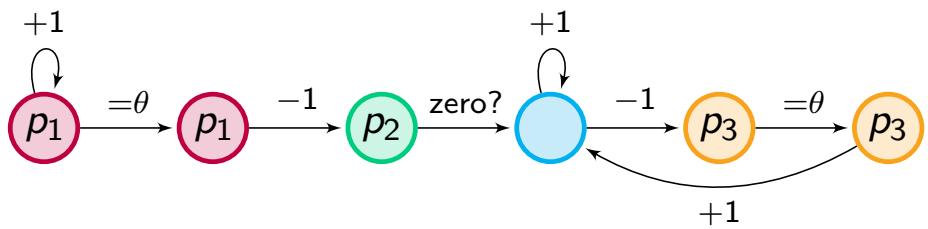
Model Checking



$\mathsf{F}(p_1 \wedge \mathsf{X}p_2) \wedge \downarrow r.(\mathsf{GF}(=r \wedge p_3))$

red. / exp.

Reachability of
OCMs with parameterized tests

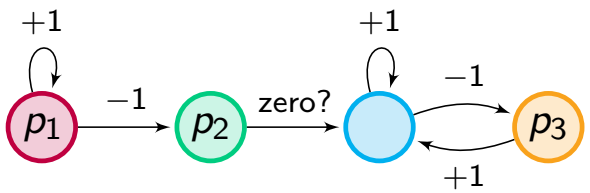


- decidability of model checking by Lechner, Mayr, Ouaknine, Pouly, Worrell, 2016
- algorithm for reachability by reduction to satisfiability of a formula in Presburger arithmetic

Flat Freeze LTL - History

- introduced by Demri and Sangnier, 2010
- main contribution:

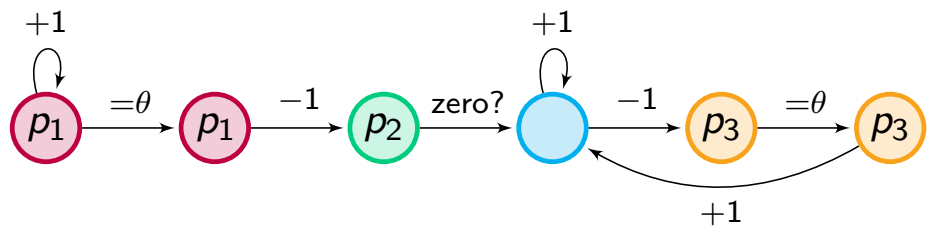
Model Checking



$\mathsf{F}(p_1 \wedge \mathsf{X}p_2) \wedge \downarrow r.(\mathsf{GF}(=r \wedge p_3))$

red. / exp.

Reachability of
OCMs with parameterized tests



- decidability of model checking by Lechner, Mayr, Ouaknine, Pouly, Worrell, 2016
- algorithm for reachability by reduction to satisfiability of a formula in Presburger arithmetic
- trivial 2NEXPTIME upper bound for model checking

Our Contribution

Theorem

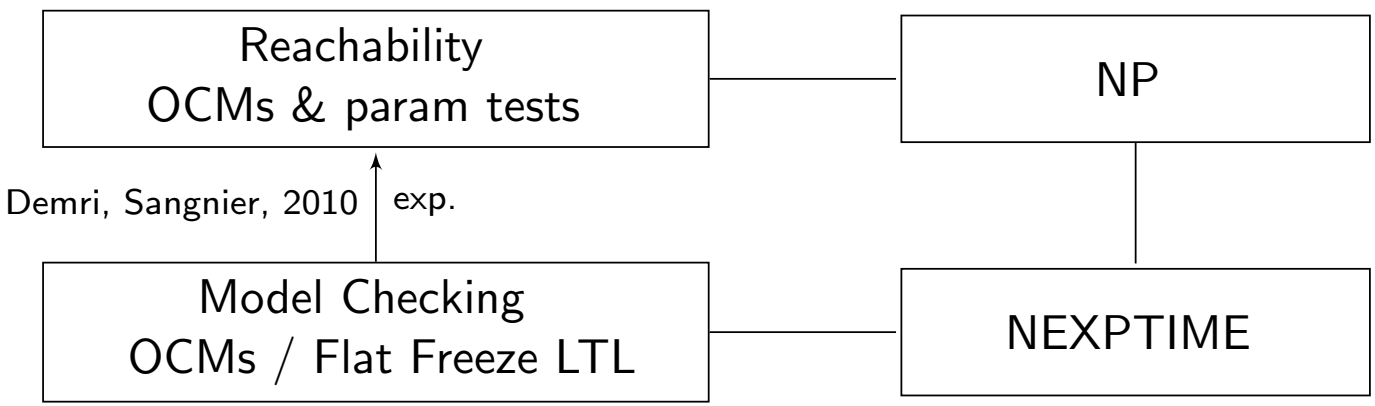
Model checking one-counter machines against formulas of Flat Freeze LTL is NEXPTIME-complete.

Our Contribution

Theorem

Model checking one-counter machines against formulas of Flat Freeze LTL is NEXPTIME-complete.

Proof outline

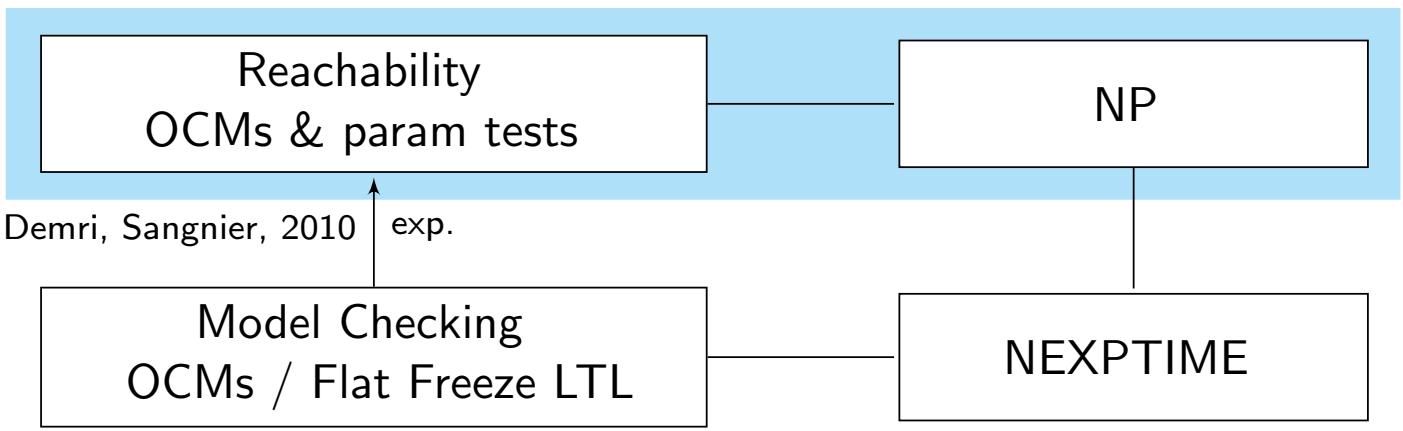


Our Contribution

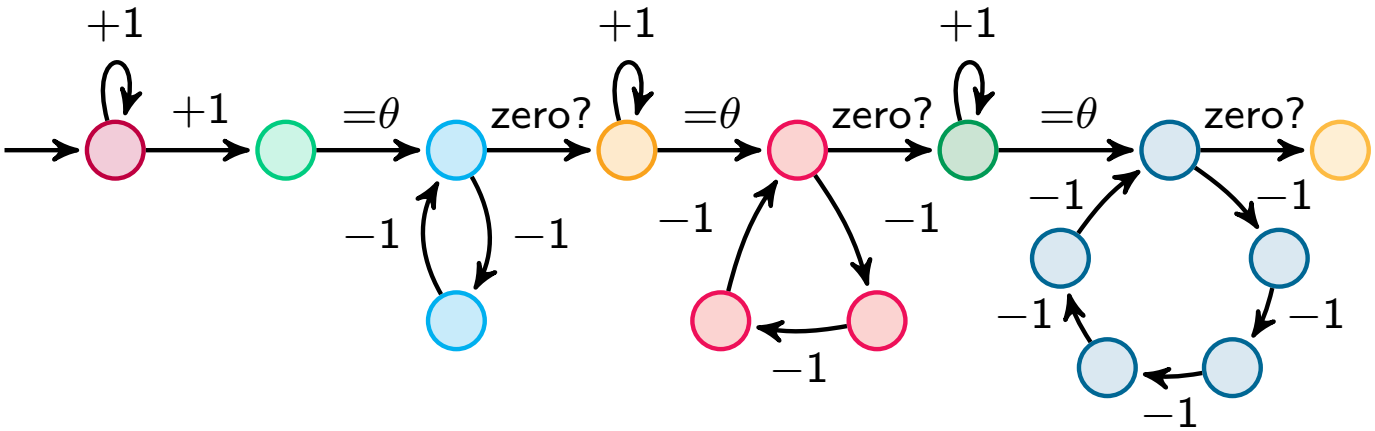
Theorem

Model checking one-counter machines against formulas of Flat Freeze LTL is NEXPTIME-complete.

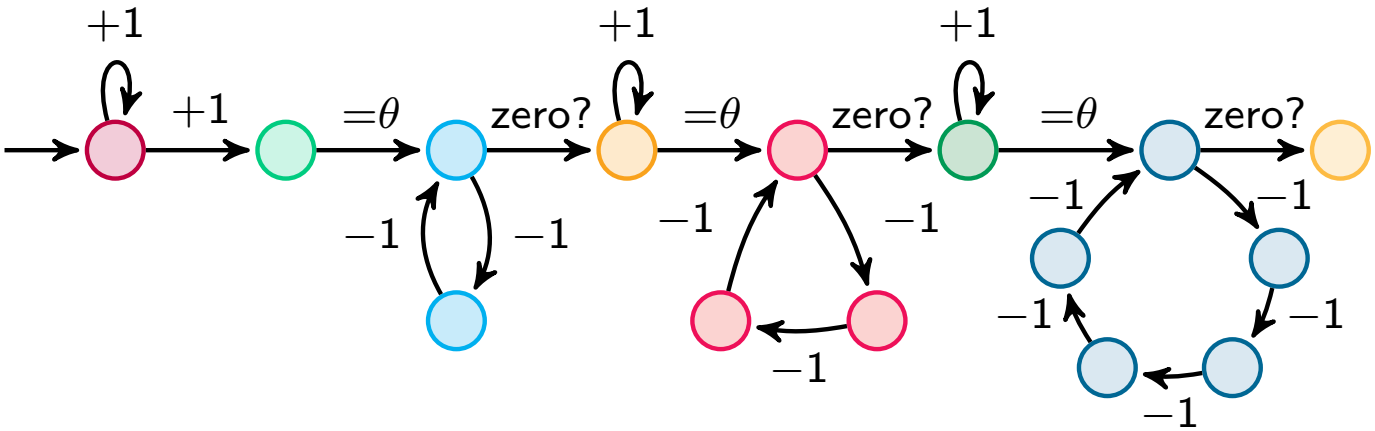
Proof outline



OCMs with Parameterized Tests

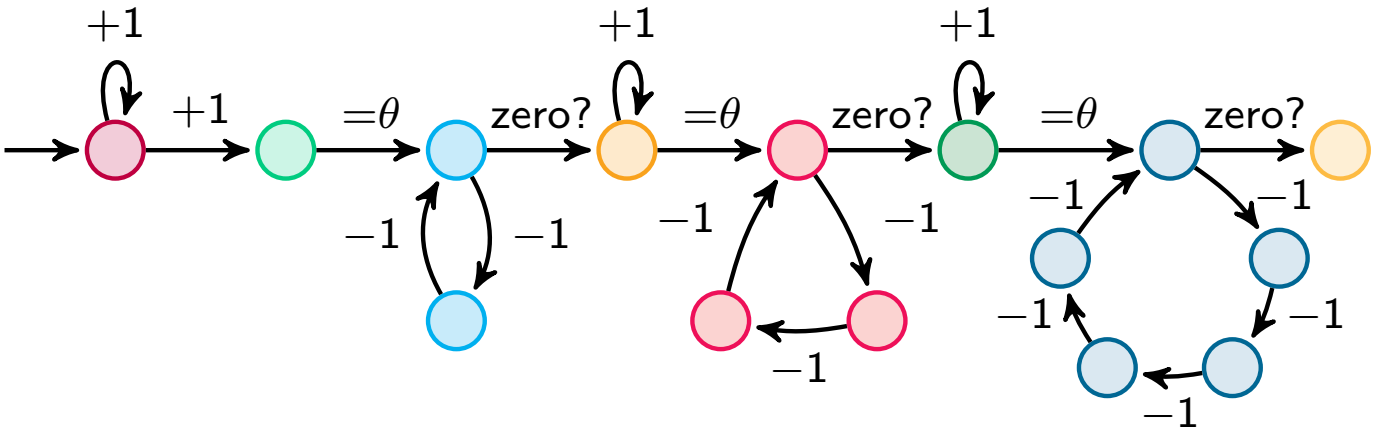


OCMs with Parameterized Tests



Parameter valuation $\gamma : \Theta \to \mathbf{N}$, e.g. $\gamma(\theta) = 30$

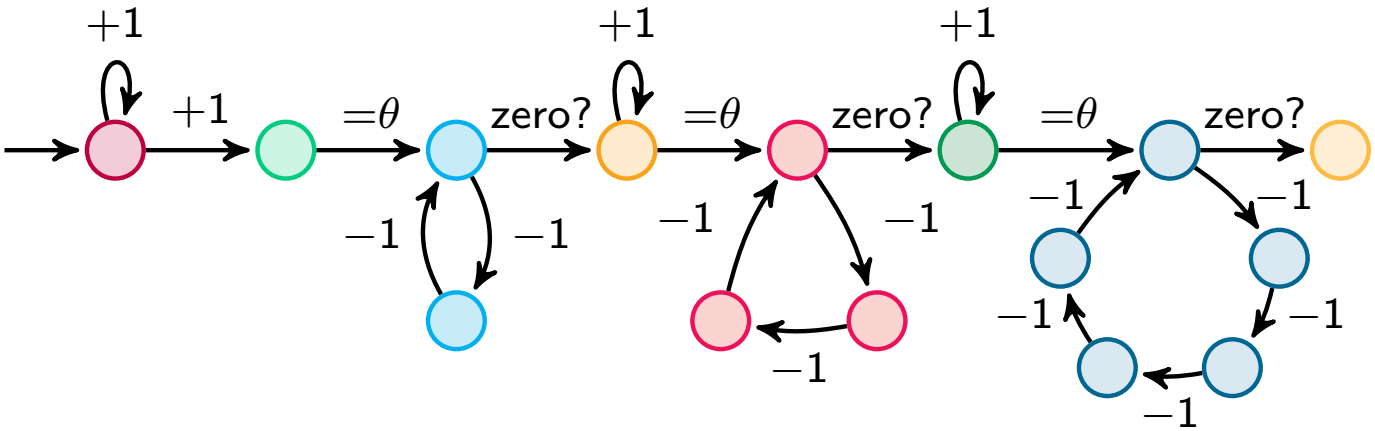
OCMs with Parameterized Tests



Parameter valuation $\gamma : \Theta \to \mathbf{N}$, e.g. $\gamma(\theta) = 30$

$0 \to_\gamma^{29} 29 \to_\gamma 30 \to_\gamma 30(\to_\gamma \to_\gamma)^{15} 0 \to_\gamma 0 \to_\gamma^{30} 30 \to_\gamma 30(\to_\gamma \to_\gamma \to_\gamma \to_\gamma)^{10} 0 \ldots$

OCMs with Parameterized Tests



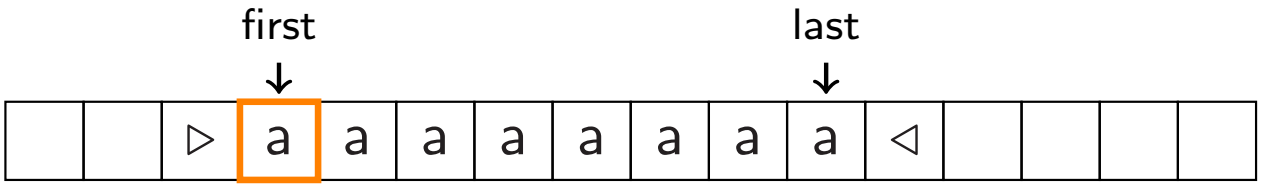
Parameter valuation $\gamma : \Theta \to \mathbf{N}$, e.g. $\gamma(\theta) = 30$

$0 \to_\gamma^{29} 29 \to_\gamma 30 \to_\gamma 30(\to_\gamma\to_\gamma)^{15}0 \to_\gamma 0 \to_\gamma^{30} 30 \to_\gamma 30(\to_\gamma\to_\gamma\to_\gamma)^{10}0\ldots$

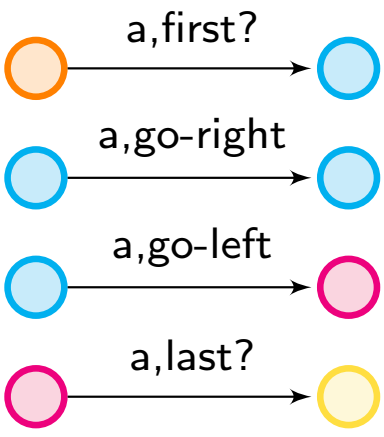
Reachability Problem

Given an OCM with parameterized tests and a target control state q, does there exist a parameter valuation γ and a γ-computation ending in q?

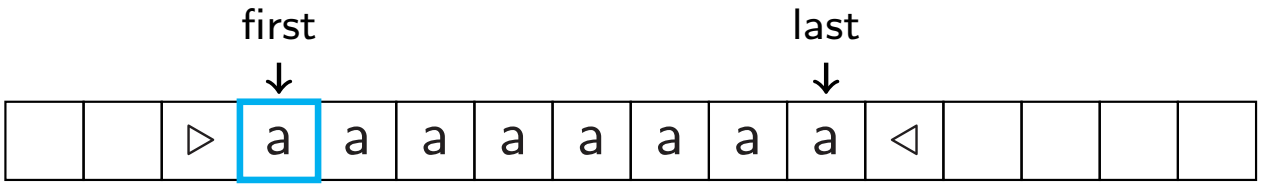
NP lower bound



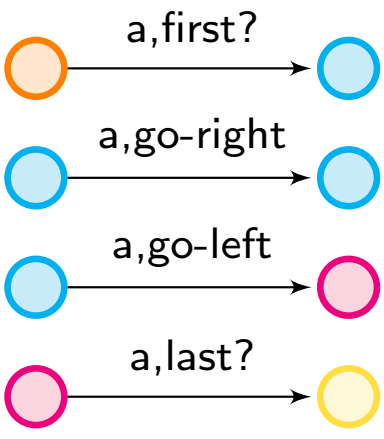
Nonemptiness
Two-Way Automata



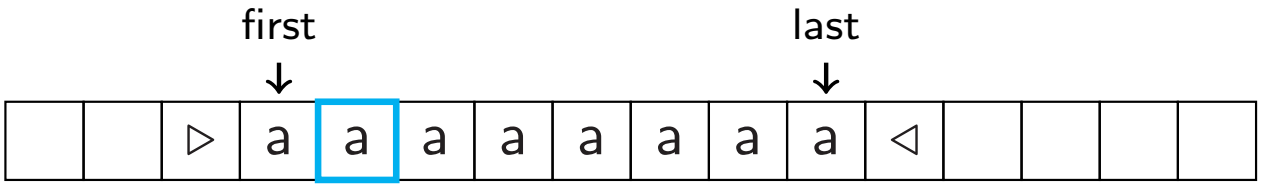
NP lower bound



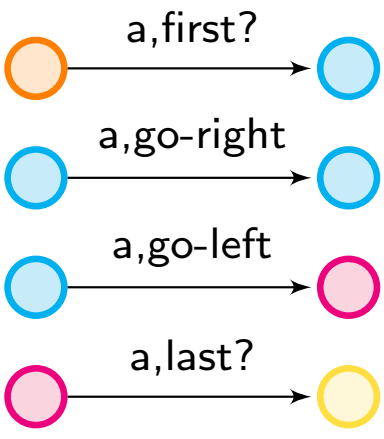
Nonemptiness
Two-Way Automata



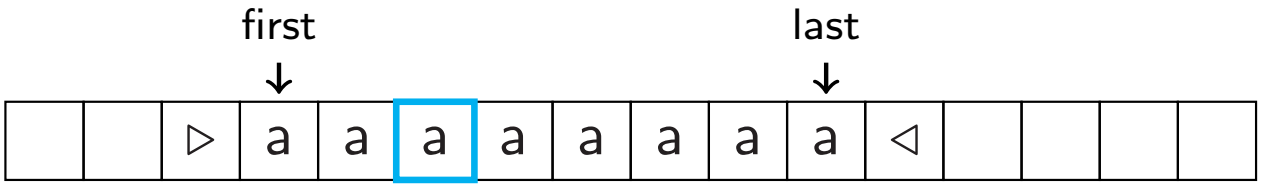
NP lower bound



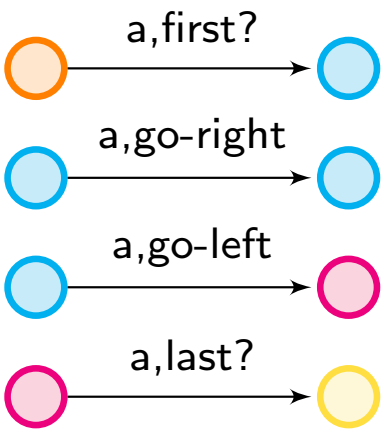
Nonemptiness
Two-Way Automata



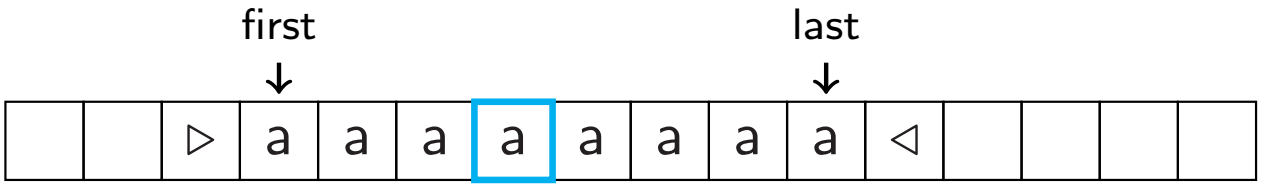
NP lower bound



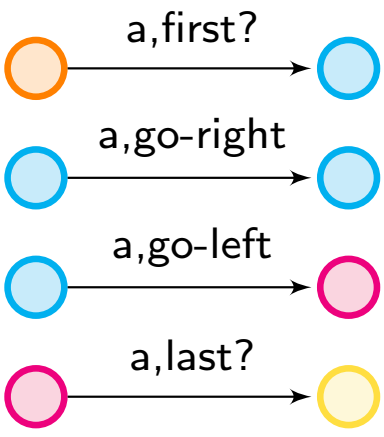
Nonemptiness
Two-Way Automata



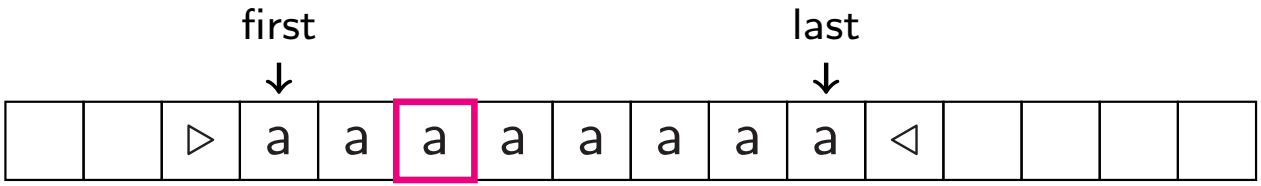
NP lower bound



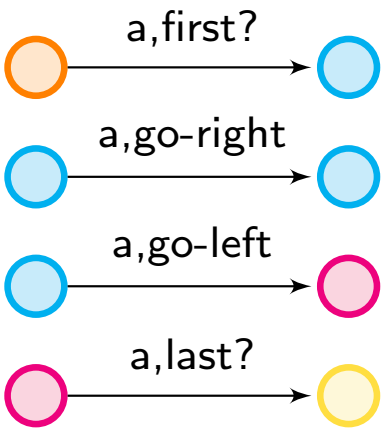
Nonemptiness
Two-Way Automata



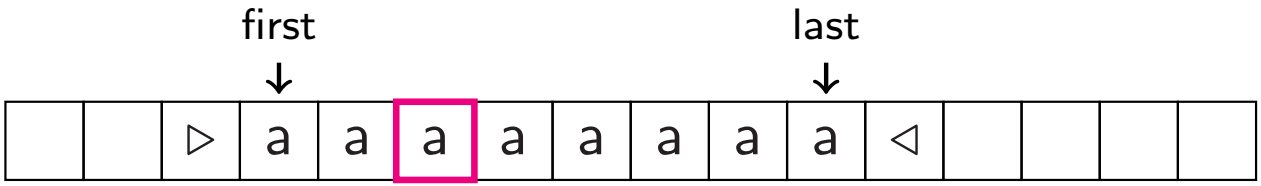
NP lower bound



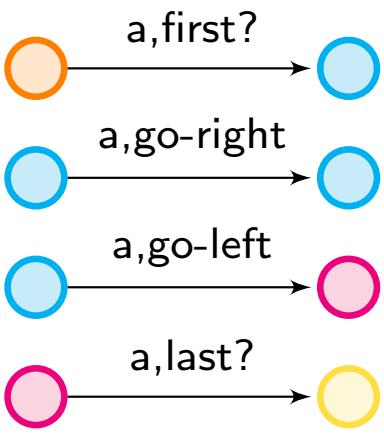
Nonemptiness
Two-Way Automata



NP lower bound



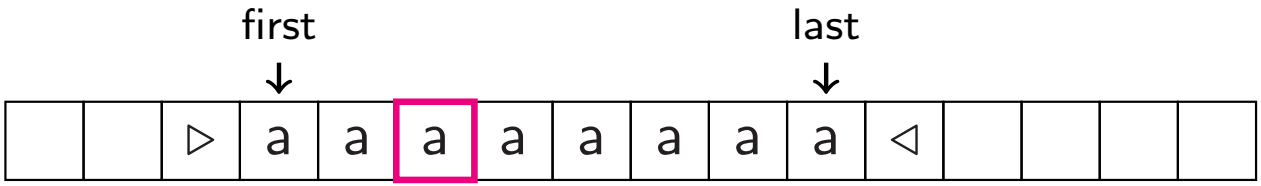
Nonemptiness
Two-Way Automata



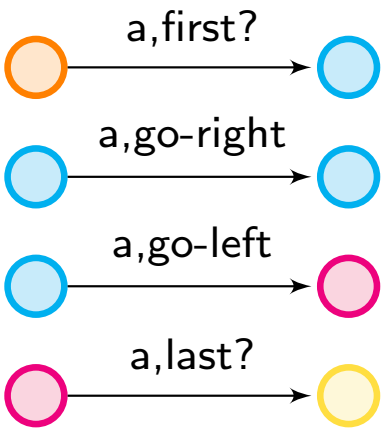
Theorem (Galil 1976)

The nonemptiness problem for two-way automata over a singleton alphabet and with two endmarkers is NP-complete.

NP lower bound



Nonemptiness
Two-Way Automata



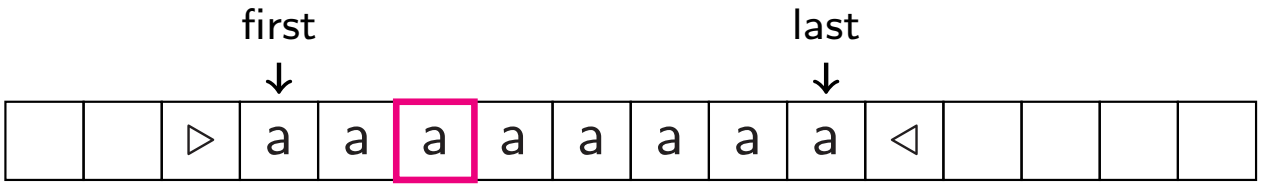
red.
⟹
poly.

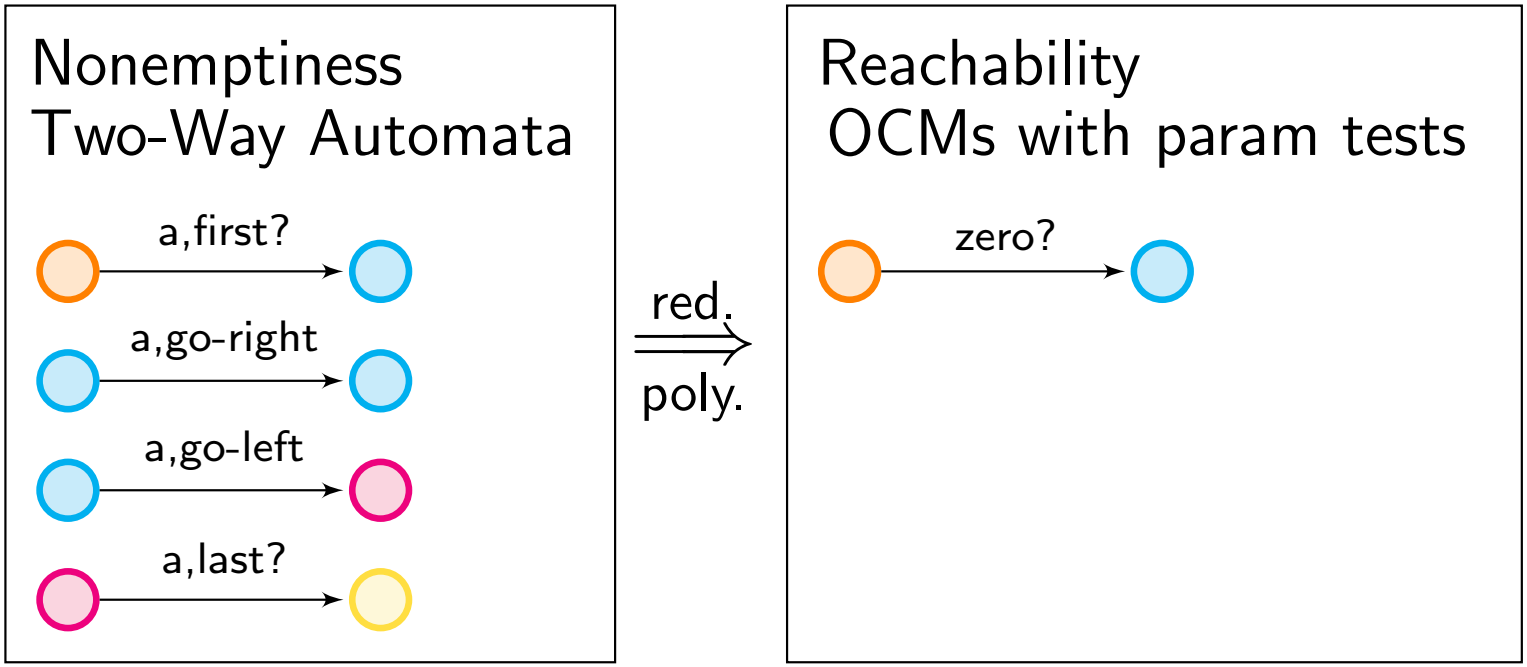
Reachability
OCMs with param tests

Theorem (Galil 1976)

The nonemptiness problem for two-way automata over a singleton alphabet and with two endmarkers is NP-complete.

NP lower bound

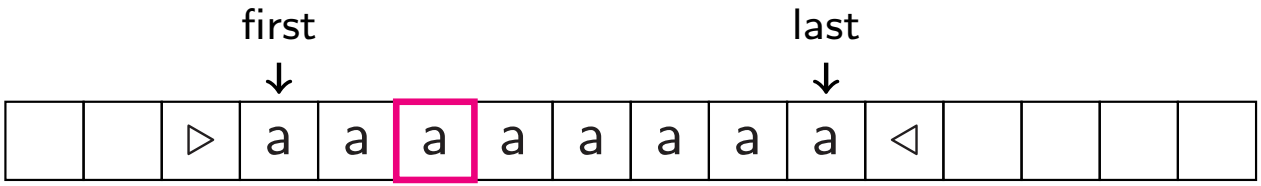


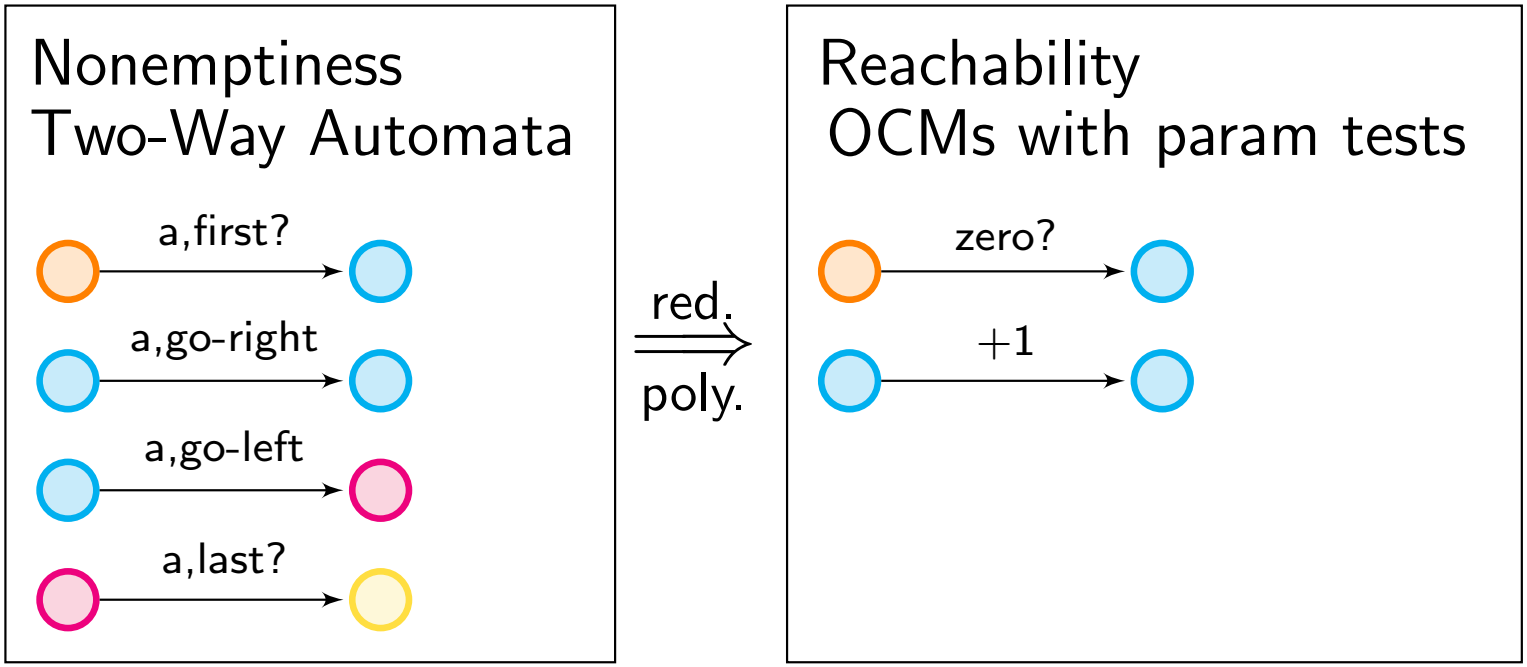


Theorem (Galil 1976)

The nonemptiness problem for two-way automata over a singleton alphabet and with two endmarkers is NP-complete.

NP lower bound

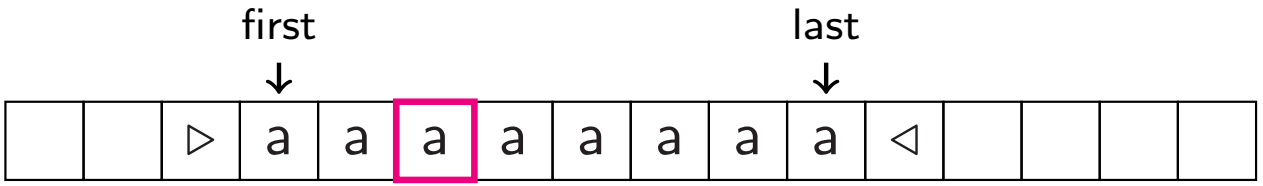


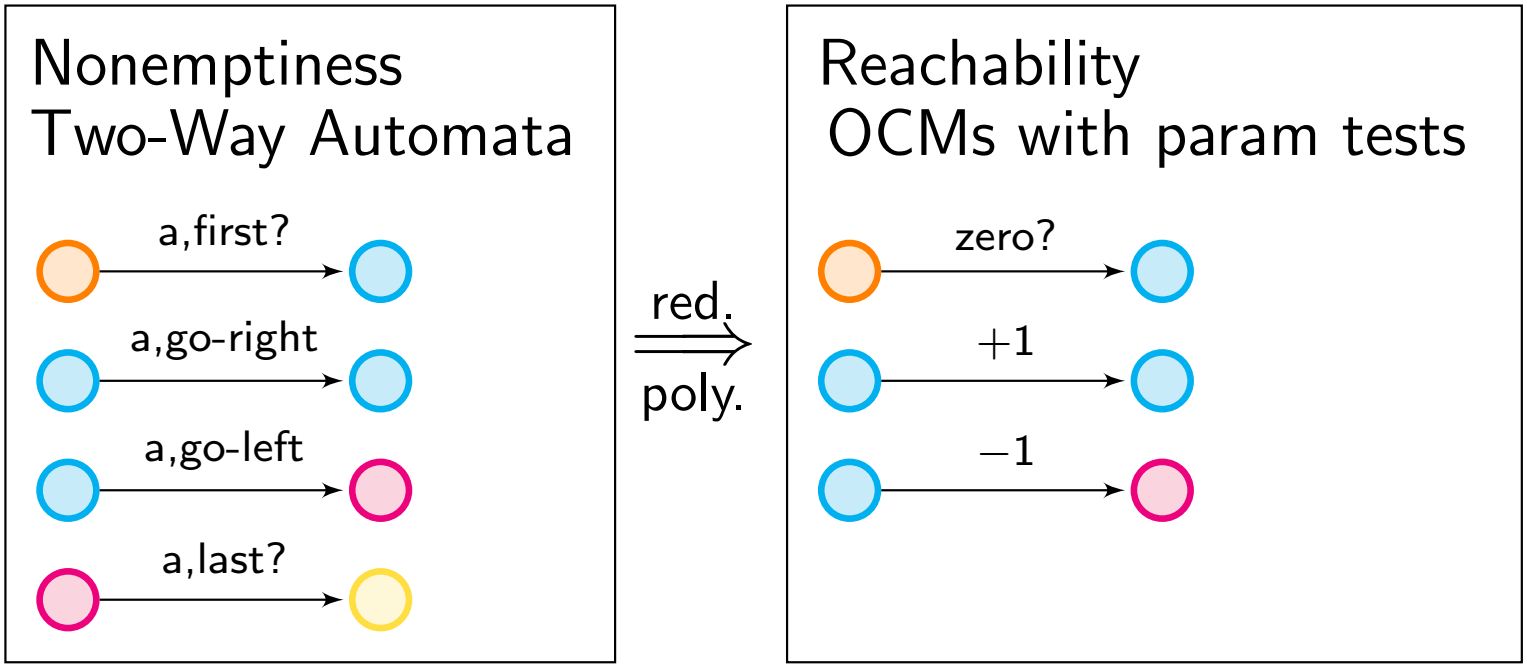


Theorem (Galil 1976)

The nonemptiness problem for two-way automata over a singleton alphabet and with two endmarkers is NP-complete.

NP lower bound

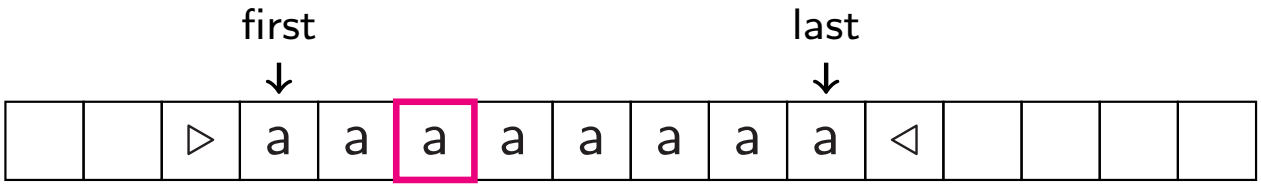


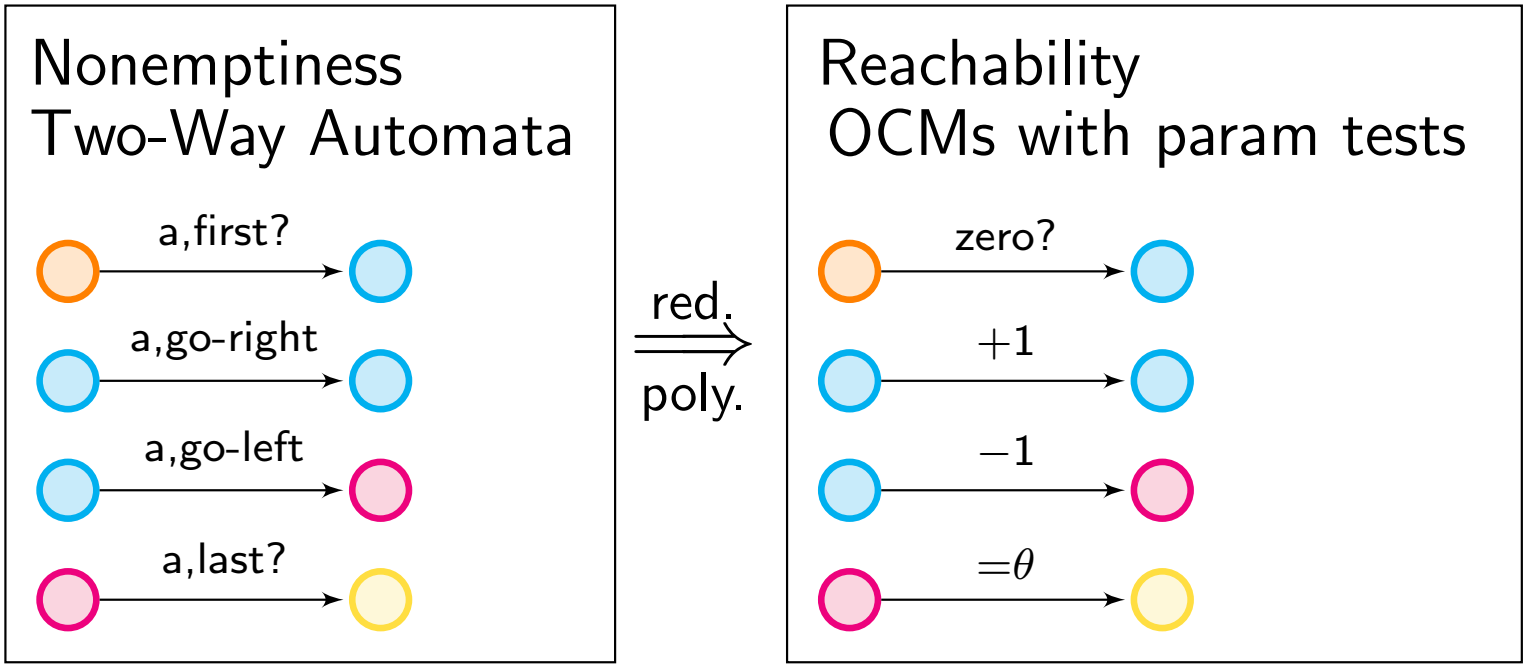


Theorem (Galil 1976)

The nonemptiness problem for two-way automata over a singleton alphabet and with two endmarkers is NP-complete.

NP lower bound

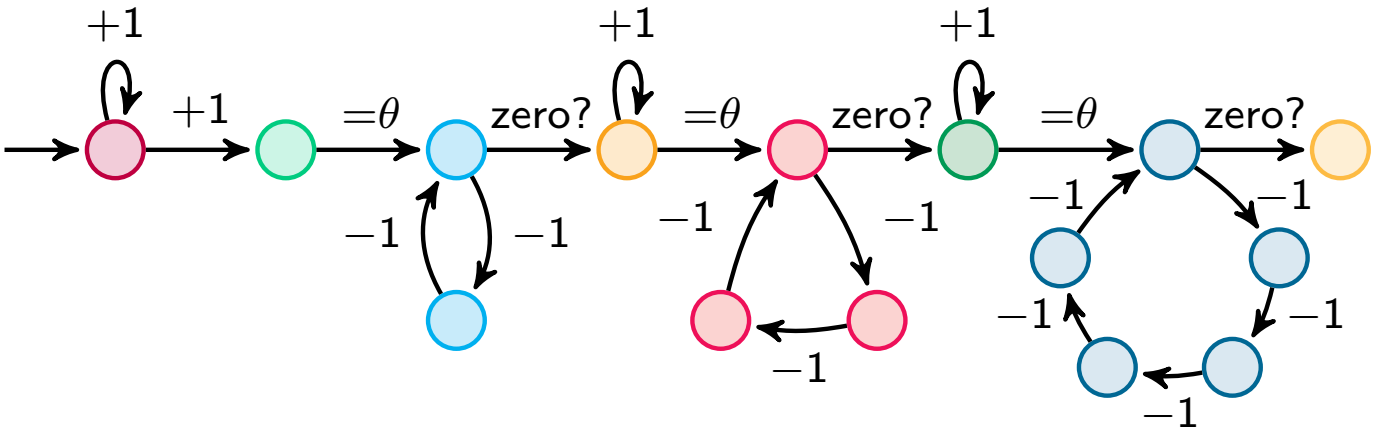




Theorem (Galil 1976)

The nonemptiness problem for two-way automata over a singleton alphabet and with two endmarkers is NP-complete.

OCMs with Parameterized Tests



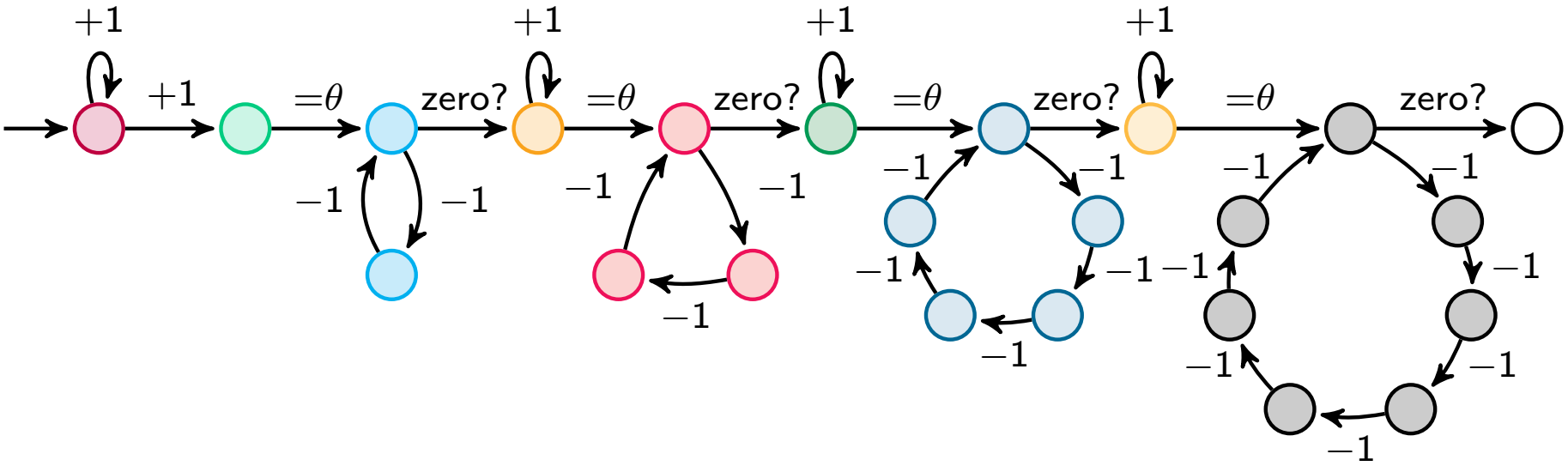
Parameter valuation $\gamma : \Theta \to \mathbf{N}$, e.g. $\gamma(\theta) = 30$

$0 \to_\gamma^{29} 29 \to_\gamma 30 \to_\gamma 30(\to_\gamma\to_\gamma)^{15}0 \to_\gamma 0 \to_\gamma^{30} 30 \to_\gamma 30(\to_\gamma\to_\gamma\to_\gamma)^{10}0\ldots$

Reachability Problem

Given an OCM with parameterized tests and a target control state q, does there exist a parameter valuation γ and a γ-computation ending in q (visiting q infinitely often)?

OCMs with Parameterized Tests



Parameter valuation $\gamma : \Theta \to \mathbf{N}$, e.g. $\gamma(\theta) = 30$

$0 \to_\gamma^{29} 29 \to_\gamma 30 \to_\gamma 30(\to_\gamma\to_\gamma)^{15}0 \to_\gamma 0 \to_\gamma^{30} 30 \to_\gamma 30(\to_\gamma\to_\gamma\to_\gamma)^{10}0\ldots$

Reachability Problem

Given an OCM with parameterized tests and a target control state q, does there exist a parameter valuation γ and a γ-computation ending in q (visiting q infinitely often)?

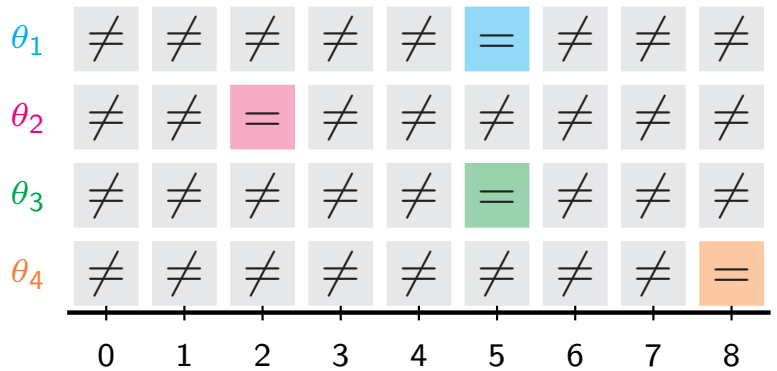
NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.

NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

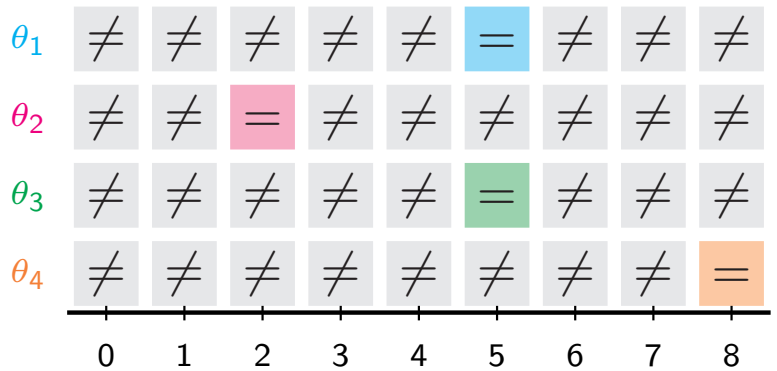
$w_\gamma =$



$\in \Sigma^*$, where $\Sigma = \{=, \neq\}^4$

NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

$w_\gamma =$ 

$\in \Sigma^*$, where $\Sigma = \{=, \neq\}^4$

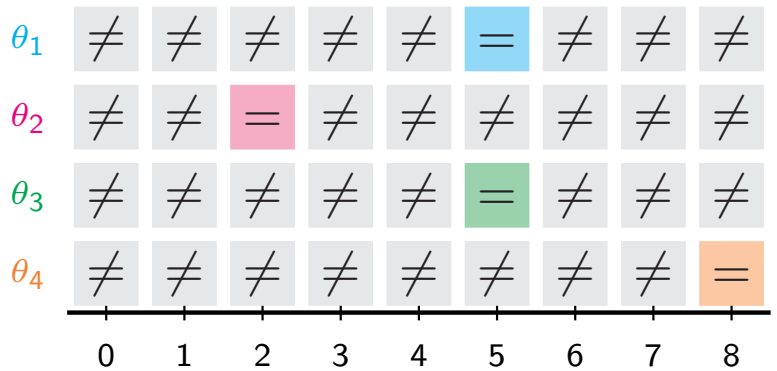
Reachability of
OCMs with param tests

$\xRightarrow[\text{poly.}]{\text{red.}}$

Nonemptiness
Two-Way Automata

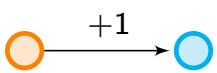
NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

$w_\gamma =$ 

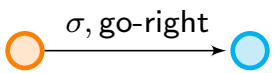
$\in \Sigma^*$, where $\Sigma = \{=, \neq\}^4$

Reachability of
OCMs with param tests



red.
poly.

Nonemptiness
Two-Way Automata



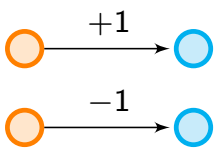
$\sigma \in \Sigma$

NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

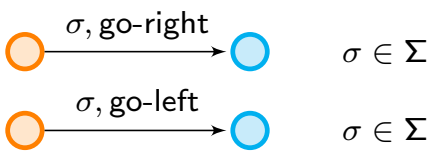
$$w_\gamma = \begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \\ \end{array} \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & = & \neq & \neq & \neq & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & \neq & \neq & \neq & = \\ \hline \end{array} \begin{array}{c} \\ \\ \\ \\ \\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \end{array} \in \Sigma^*, \text{ where } \Sigma = \{=, \neq\}^4$$

Reachability of
OCMs with param tests



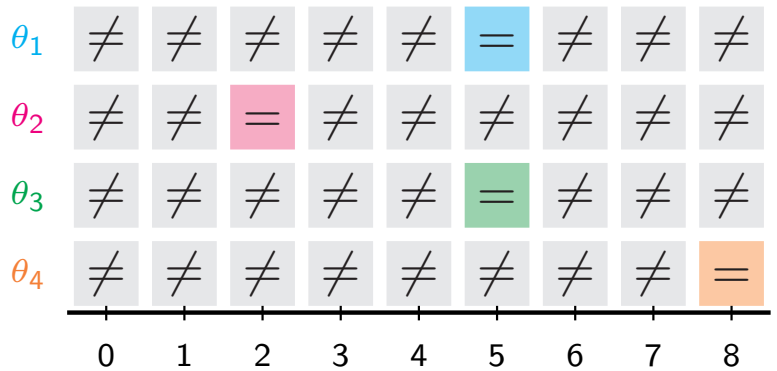
red.
poly.

Nonemptiness
Two-Way Automata



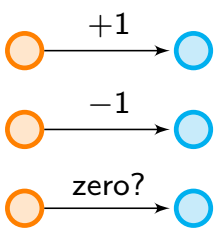
NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation. The following "parameter word" w_γ encodes γ.

$w_\gamma =$ 

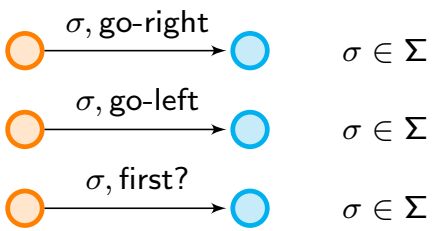
$\in \Sigma^*$, where $\Sigma = \{=, \neq\}^4$

Reachability of OCMs with param tests



red. / poly.

Nonemptiness Two-Way Automata

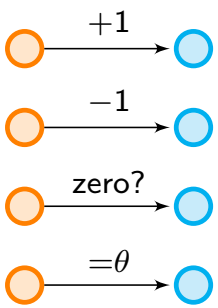


NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

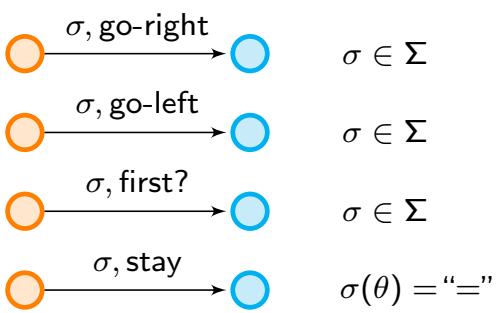
$$w_\gamma = \begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \\ \end{array} \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & = & \neq & \neq & \neq & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & \neq & \neq & \neq & = \\ \hline \end{array} \begin{array}{c} \\ \\ \\ \\ \\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \end{array} \in \Sigma^*, \text{ where } \Sigma = \{=, \neq\}^4$$

Reachability of OCMs with param tests



red. ⟹ poly.

Nonemptiness Two-Way Automata

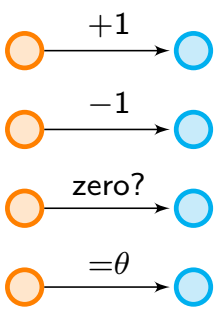


NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

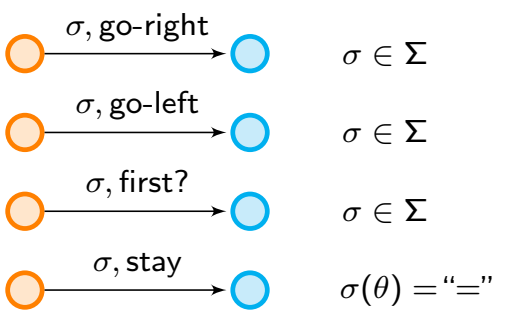
$$w_\gamma = \begin{array}{c|ccccccccc} \theta_1 & \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \theta_2 & \neq & \neq & = & \neq & \neq & \neq & \neq & \neq & \neq \\ \theta_3 & \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \theta_4 & \neq & \neq & \neq & \neq & \neq & \neq & \neq & \neq & = \\ \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \in \Sigma^*, \text{ where } \Sigma = \{=, \neq\}^4$$

Reachability of OCMs with param tests



red. poly.

Nonemptiness Two-Way Automata



red. exp.

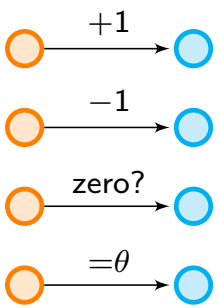
Nonemptiness Finite Automata

NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

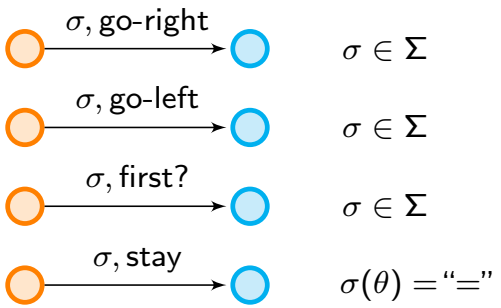
$$w_\gamma = \begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \\ \end{array} \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & = & \neq & \neq & \neq & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & \neq & \neq & \neq & = \\ \hline \end{array} \begin{array}{c} \\ \\ \\ \\ \\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \end{array} \in \Sigma^*, \text{ where } \Sigma = \{=, \neq\}^4$$

Reachability of OCMs with param tests



red. ⟹ poly.

Nonemptiness Two-Way Automata



red. ⟹ exp.

Nonemptiness Finite Automata

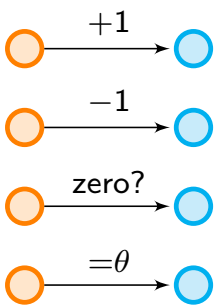
length of smallest witness for nonemptiness is exponential in the size of the OCM

NP upper bound (1)

Let $\gamma = \{\theta_1 \mapsto 5, \theta_2 \mapsto 2, \theta_3 \mapsto 5, \theta_4 \mapsto 8\}$ be a parameter valuation.
The following "parameter word" w_γ encodes γ.

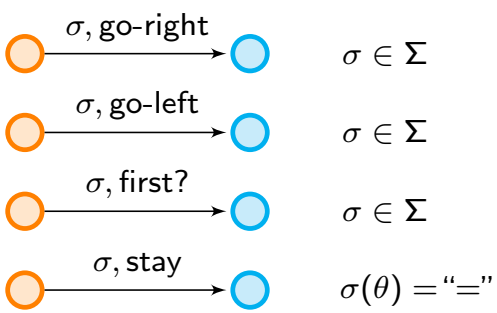
$$w_\gamma = \begin{array}{c} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \\ \end{array} \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & = & \neq & \neq & \neq & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & = & \neq & \neq & \neq \\ \hline \neq & \neq & \neq & \neq & \neq & \neq & \neq & \neq & = \\ \hline \end{array} \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \in \Sigma^*, \text{ where } \Sigma = \{=, \neq\}^4$$

Reachability of
OCMs with param tests



red.
poly.

Nonemptiness
Two-Way Automata



red.
exp.

Nonemptiness
Finite Automata

length of smallest
witness for nonemptiness
is exponential in
the size of the OCM

↷ parameter valuation
is bounded exponentially

NP upper bound (2)

Given an OCM with parameterized tests $\mathcal{A}$ and a target control state q,

- if q is reachable, then it is reachable under some parameter valuation γ such that $\gamma(\theta)$ is bounded exponentially in the size of $\mathcal{A}$, for every θ

NP upper bound (2)

Given an OCM with parameterized tests $\mathcal{A}$ and a target control state q,

- if q is reachable, then it is reachable under some parameter valuation γ such that $\gamma(\theta)$ is bounded exponentially in the size of $\mathcal{A}$, for every θ
- $\curvearrowright$ if q is reachable, then it is reachable even if the counter values along the computation are bounded exponentially in the size of $\mathcal{A}$

NP upper bound (2)

Given an OCM with parameterized tests $\mathcal{A}$ and a target control state q,

- if q is reachable, then it is reachable under some parameter valuation γ such that $\gamma(\theta)$ is bounded exponentially in the size of $\mathcal{A}$, for every θ
- $\curvearrowright$ if q is reachable, then it is reachable even if the counter values along the computation are bounded exponentially in the size of $\mathcal{A}$
- $\curvearrowright$ guess a parameter valuation and a bound on the counter in binary representation, and determine in polynomial time whether this constitutes a computation reaching q

NP upper bound (2)

Given an OCM with parameterized tests $\mathcal{A}$ and a target control state q,

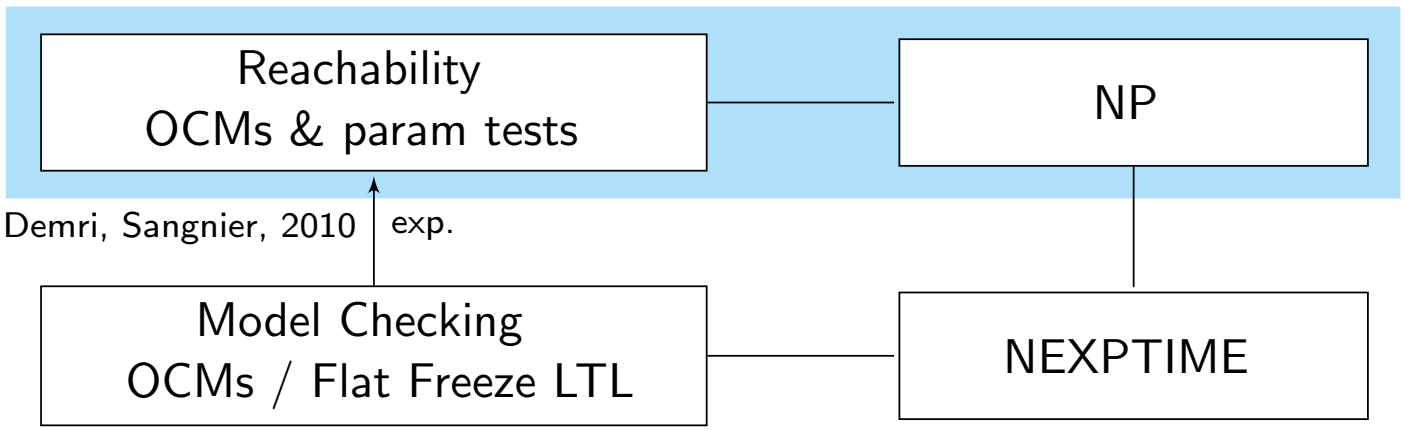
- if q is reachable, then it is reachable under some parameter valuation γ such that $\gamma(\theta)$ is bounded exponentially in the size of $\mathcal{A}$, for every θ
- $\curvearrowright$ if q is reachable, then it is reachable even if the counter values along the computation are bounded exponentially in the size of $\mathcal{A}$
- $\curvearrowright$ guess a parameter valuation and a bound on the counter in binary representation, and determine in polynomial time whether this constitutes a computation reaching q (proved using techniques from Galil 1976)

Our Contribution

Theorem

Model checking one-counter machines against formulas of Flat Freeze LTL is NEXPTIME-complete.

Proof outline

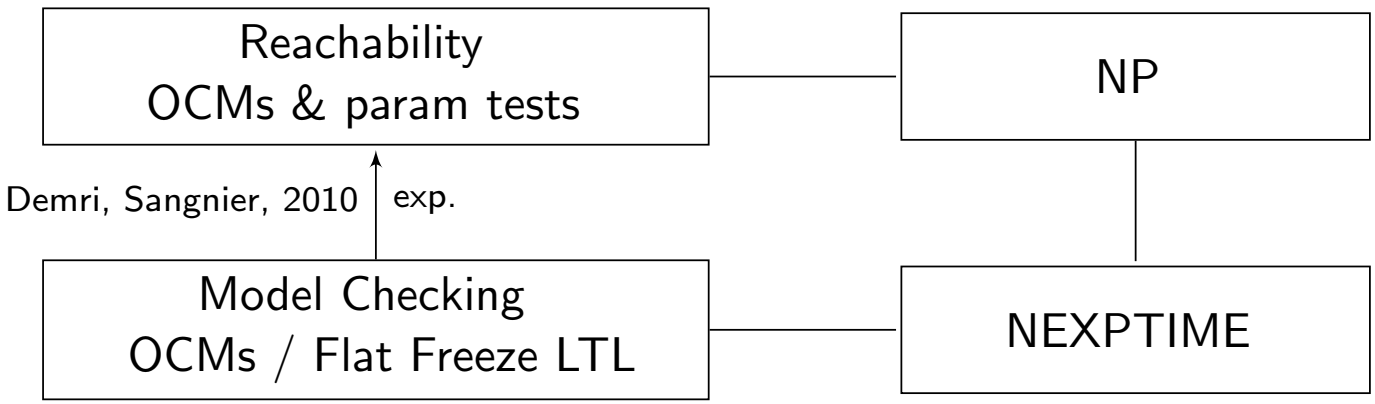


Our Contribution

Theorem

Model checking one-counter machines against formulas of Flat Freeze LTL is NEXPTIME-complete.

Proof outline

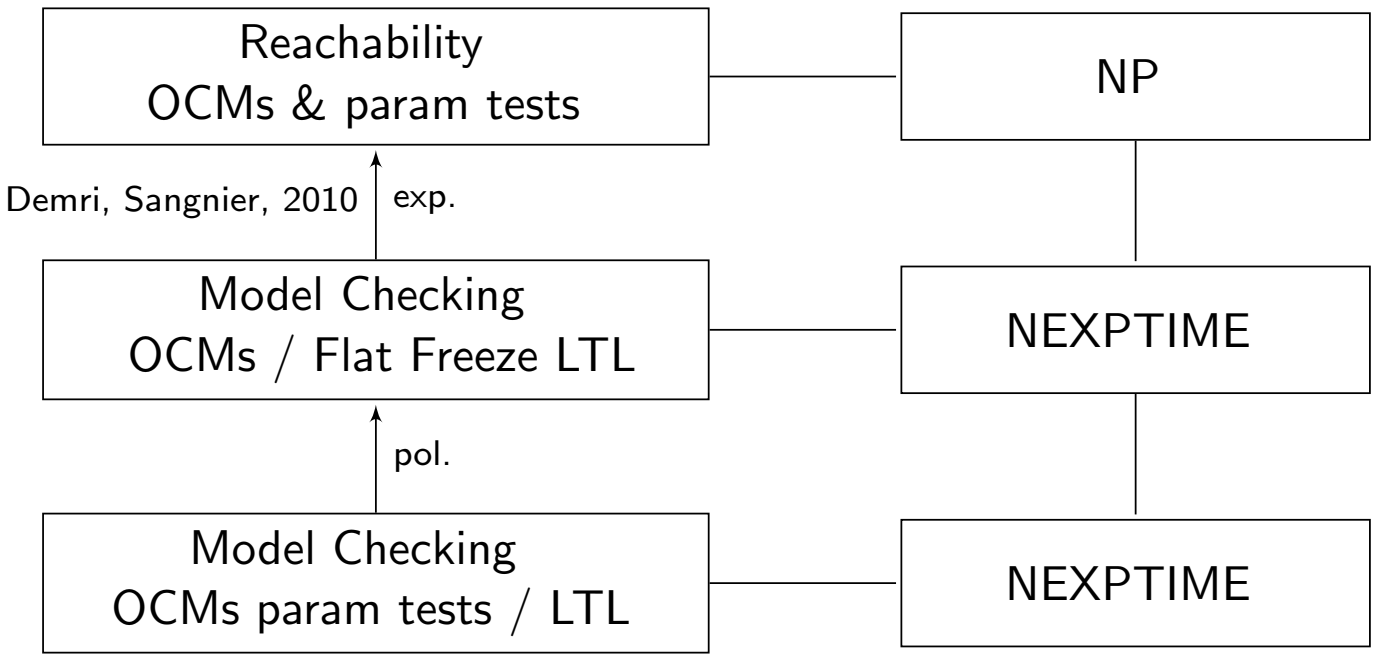


Our Contribution

Theorem

Model checking one-counter machines against formulas of Flat Freeze LTL is NEXPTIME-complete.

Proof outline



Summary and Open Problems

Summary and Open Problems

- We re-used a tight link between one-counter machines and two-way automata

Summary and Open Problems

- We re-used a tight link between one-counter machines and two-way automata
- All proofs also hold for
    - the infinite case,

Summary and Open Problems

- We re-used a tight link between one-counter machines and two-way automata
- All proofs also hold for
    - the infinite case,
    - tests of the form $<r$ and $>r$, and

Summary and Open Problems

- We re-used a tight link between one-counter machines and two-way automata
- All proofs also hold for
    - the infinite case,
    - tests of the form $<r$ and $>r$, and
    - succinct (binary encoded) counter updates

Summary and Open Problems

- We re-used a tight link between one-counter machines and two-way automata
- All proofs also hold for
    - the infinite case,
    - tests of the form $<r$ and $>r$, and
    - succinct (binary encoded) counter updates

- One-counter machines with parameterized tests and updates

Summary and Open Problems

- We re-used a tight link between one-counter machines and two-way automata
- All proofs also hold for
    - the infinite case,
    - tests of the form $<r$ and $>r$, and
    - succinct (binary encoded) counter updates

- One-counter machines with parameterized tests and updates
- Parameterized timed automata (single clock case), where only a non-trivial 2NEXPTIME upper bound is known