Reachability problem for polynomial iteration is PSPACE-complete

Reino Niskanen

Department of Computer Science
University of Liverpool, UK

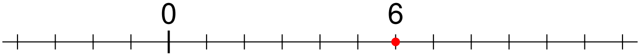
11th International Workshop on Reachability Problems

Introduction

Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

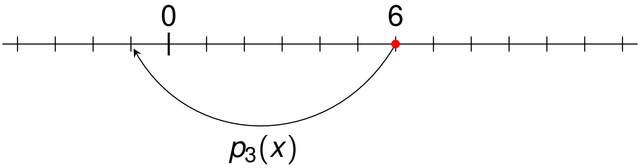
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

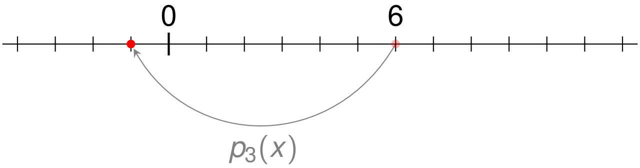
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

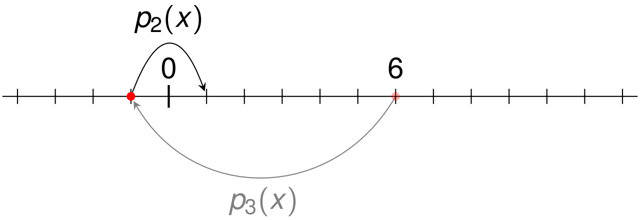
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

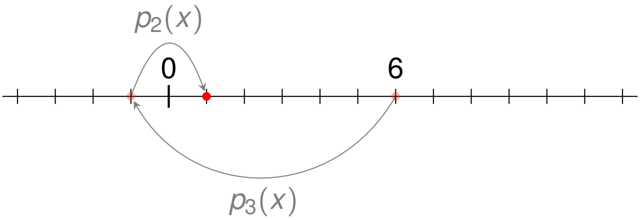
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

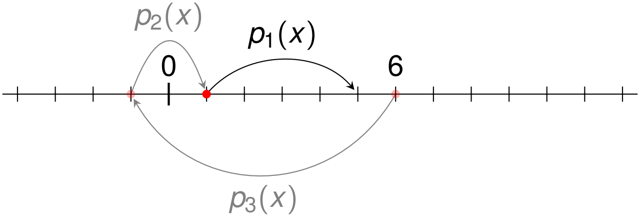
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

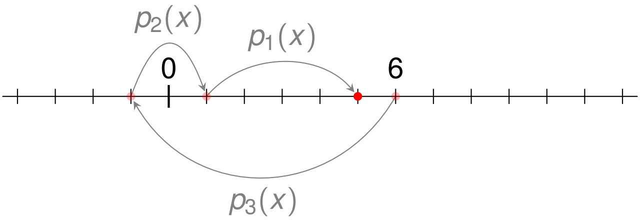
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

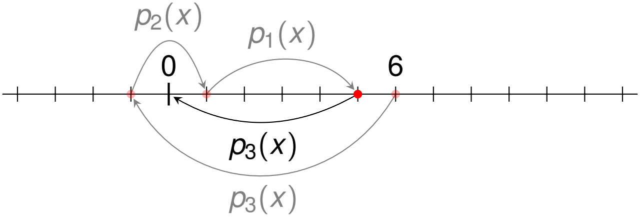
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

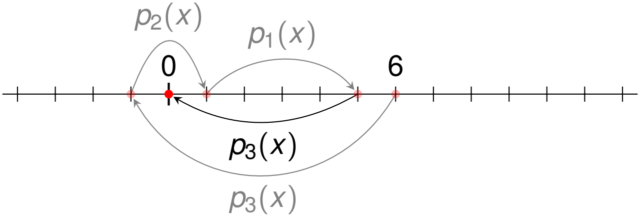
Can we iterate $x = 6$ to reach 0?



Polynomial iteration

$$p_1(x) = x^2 + x + 3$$
$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$
$$p_3(x) = -x + 5$$

Can we iterate $x = 6$ to reach 0?



Polynomial iteration

How much space is needed?

$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$

Polynomial iteration

How much space is needed?

$$p_2(x) = x^4 + 2x^3 + 3x^2 + 2x + 1$$

A lot..

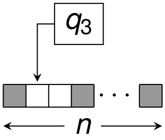
$$6 \mapsto 1849 \mapsto 11700853263801$$

The representation grows exponentially.

Definitions

Linear bounded automata

- Linear bounded automata is a Turing machine with a finite tape whose length is bounded by a linear function of the size of the input.
- A configuration is $[q, i, w]$, where $q \in Q$, i is the position of the head, $w \in \{0, 1\}^n$ is the word written on the tape.
- The reachability problem: $[q_0, 1, 0^n] \to^* [q_f, 1, 0^n]$?

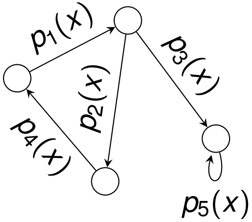


Theorem

The reachability problem for LBA *is* PSPACE*-complete.*

Polynomial register machines

- Introduced by Finkel, Göller and Haase in MFCS'13
- A PRM consists of a graph (V, E) labelled by polynomials in $\mathbb{Z}[x]$.
- A configuration is $[s, z] \in V \times \mathbb{Z}$.
- $[s, z]$ yields $[s', y]$ if $(s, p(x), s') \in E$ such that $p(z) = y$.
- The reachability problem: $[s_0, 0] \to^* [s_f, 0]$?



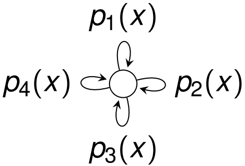
Theorem (FGH 2013)

The reachability problem for PRM *is* PSPACE*-complete.*

Polynomial iteration

- Can be seen as stateless PRMs.
- $\mathcal{P} = \{p_1(x), p_2(x), \ldots, p_n(x)\} \subseteq \mathbb{Z}[x]$.
- The reachability problem: Does there exist a finite sequence $p_{i_1}(x), p_{i_2}(x), \ldots, p_{i_j}(x)$ that maps x_0 to x_f, i.e., whether

$$p_{i_j}(p_{i_{j-1}}(\cdots p_{i_2}(p_{i_1}(x_0))\cdots)) = x_f.$$



Theorem

The reachability problem for polynomial iteration is PSPACE*-complete.*

Polynomial iteration

Upper bound

Lemma

The reachability problem for polynomial iteration is PSPACE*.*

Proof.

The reachability problem is PSPACE even for machines with states. □

Upper bound

Lemma

The reachability problem for polynomial iteration is PSPACE.

Idea of Proof

- For almost all polynomials $p(x)$, there exists a bound b, such that for any $|y| > b$, $|p(y)| \geq 2|y|$.

Upper bound

Lemma

The reachability problem for polynomial iteration is PSPACE.

Idea of Proof

- For almost all polynomials $p(x)$, there exists a bound b, such that for any $|y| > b$, $|p(y)| \geq 2|y|$.
- Only polynomials $\pm x + a$, for some $a \in \mathbb{Z}$, do not have this bound. Their behaviour can be simulated by a 1-VASS, for which the reachability problem is in NP.

Upper bound

Lemma

The reachability problem for polynomial iteration is PSPACE.

Idea of Proof

- For almost all polynomials $p(x)$, there exists a bound b, such that for any $|y| > b$, $|p(y)| \geq 2|y|$.
- Only polynomials $\pm x + a$, for some $a \in \mathbb{Z}$, do not have this bound. Their behaviour can be simulated by a 1-VASS, for which the reachability problem is in NP.
- Moreover, it can be simulated in polynomial space, to which values inside $[-b, b]$ the polynomials $\pm x + a$ return to.

Lower bound

Lemma

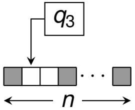
The reachability problem for polynomial iteration is PSPACE*-hard.*

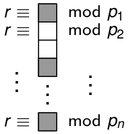
Idea of Proof

Follow the proof for PRM by reducing from the reachability of LBA. Additionally, encode states and state transitions as polynomials.

Ingredients of the reduction of LBA to PRM

Let $p_1, \ldots, p_n \in \textsc{Prime}$. We consider an integer x as a residue class $r \mod p_1 \cdots p_n$.
The tape word $w \in \{0,1\}^n$ is encoded as an integer r satisfying $r \equiv w_i \mod p_i$ for each $i = 1, \ldots, n$.





Ingredients of the reduction of LBA to PRM

Let $p_1, \ldots, p_n \in \textsc{Prime}$. We consider an integer x as a residue class $r \mod p_1 \cdots p_n$.
The tape word $w \in \{0,1\}^n$ is encoded as an integer r satisfying $r \equiv w_i \mod p_i$ for each $i = 1, \ldots, n$.

We only consider integers that are solutions to

$$\begin{aligned} r &\equiv b_1 \mod p_1 \\ &\vdots \\ r &\equiv b_n \mod p_n, \end{aligned} \qquad \text{where } b_i \in \{0,1,2\}.$$

Ingredients of the reduction of LBA to PRM

Polynomials that *locally* modify residue classes.

- FLIP_i to switch between $r \equiv 0 \mod p_i$ and $r' \equiv 1 \mod p_i$
- EQZERO_i to check that $r \equiv 0 \mod p_i$
- EQONE_i to check that $r \equiv 1 \mod p_i$.

While the other congruences remain untouched.

The update polynomials

if $r \equiv 0 \mod p_i$:

$$\mathrm{FLIP}_i(r) \equiv \begin{cases} 1 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 1 \mod p_i$:

$$\mathrm{FLIP}_i(r) \equiv \begin{cases} 0 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 2 \mod p_i$:

$$\mathrm{FLIP}_i(r) \equiv \begin{cases} 2 & \mod p_i \\ r & \mod p_j. \end{cases}$$

The update polynomials

if $r \equiv 0 \mod p_i$:
$$\text{FLIP}_i(r) \equiv \begin{cases} 1 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 1 \mod p_i$:
$$\text{FLIP}_i(r) \equiv \begin{cases} 0 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 2 \mod p_i$:
$$\text{FLIP}_i(r) \equiv \begin{cases} 2 & \mod p_i \\ r & \mod p_j. \end{cases}$$

is realised by

$$p_{flip,i}(x) = a_2' x^2 + a_1' x + a_0'$$

$$\begin{cases} a_2' & \equiv 3^{\frac{p_i+1}{2}} \mod p_i \\ a_2' & \equiv 0 \mod p_j \end{cases} \quad \begin{cases} a_1' \equiv -5^{\frac{p_i+1}{2}} \mod p_i \\ a_1' \equiv 1 \mod p_j \end{cases} \quad \begin{cases} a_0' \equiv 1 \mod p_i \\ a_0' \equiv 0 \mod p_j \end{cases}$$

The update polynomials

if $r \equiv 0 \mod p_i$:

$$\mathrm{EQZERO}_i(r) \equiv \begin{cases} 0 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 1, 2 \mod p_i$:

$$\mathrm{EQZERO}_i(r) \equiv \begin{cases} 2 & \mod p_i \\ r & \mod p_j \end{cases}$$

The update polynomials

if $r \equiv 0 \mod p_i$:

$$\text{EQZERO}_i(r) \equiv \begin{cases} 0 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 1, 2 \mod p_i$:

$$\text{EQZERO}_i(r) \equiv \begin{cases} 2 & \mod p_i \\ r & \mod p_j \end{cases}$$

is realised by

$$p_{eqzero,i}(x) = a'_2 x^2 + a'_1 x + a'_0$$

$$\begin{cases} a'_2 \equiv -1 \mod p_i \\ a'_2 \equiv 0 \mod p_j \end{cases} \quad \begin{cases} a'_1 \equiv 3 \mod p_i \\ a'_1 \equiv 1 \mod p_j \end{cases} \quad \begin{cases} a'_0 \equiv 0 \mod p_i \\ a'_0 \equiv 0 \mod p_j \end{cases}$$

The update polynomials

if $r \equiv 1 \mod p_i$:

$$\text{EQONE}_i(r) \equiv \begin{cases} 1 & \mod p_i \\ r & \mod p_j \end{cases}$$

if $r \equiv 0, 2 \mod p_i$:

$$\text{EQONE}_i(r) \equiv \begin{cases} 2 & \mod p_i \\ r & \mod p_j \end{cases}$$

The update polynomials

if $r \equiv 1 \mod p_i$:

$$\text{EQONE}_i(r) \equiv \begin{cases} 1 \mod p_i \\ r \mod p_j \end{cases}$$

if $r \equiv 0, 2 \mod p_i$:

$$\text{EQONE}_i(r) \equiv \begin{cases} 2 \mod p_i \\ r \mod p_j \end{cases}$$

is realised by

$$p_{eqone,i}(x) = a'_2 x^2 + a'_1 x + a'_0$$

$$\begin{cases} a'_2 \equiv 1 \mod p_i \\ a'_2 \equiv 0 \mod p_j \end{cases} \quad \begin{cases} a'_1 \equiv -2 \mod p_i \\ a'_1 \equiv 1 \mod p_j \end{cases} \quad \begin{cases} a'_0 \equiv 2 \mod p_i \\ a'_0 \equiv 0 \mod p_j \end{cases}$$

Ingredients of the reduction of LBA to PRM

States of PRM contain information on the state of LBA, position of the head, and which symbol it is reading.

Ingredients of the reduction of LBA to PRM

States of PRM contain information on the state of LBA, position of the head, and which symbol it is reading.

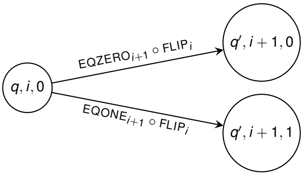
Connect the states using correct FLIP_i, EQZERO_i and EQONE_i moves. The machine guesses and verifies the symbols that will be read next.

Ingredients of the reduction of LBA to PRM

States of PRM contain information on the state of LBA, position of the head, and which symbol it is reading.

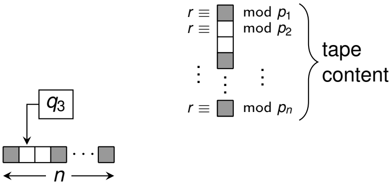
Connect the states using correct FLIP_i, EQZERO_i and EQONE_i moves. The machine guesses and verifies the symbols that will be read next.

For a move $\delta(q, 0) = (q', 1, R)$ of the LBA, the states and transitions of the PRM (for each i) are:



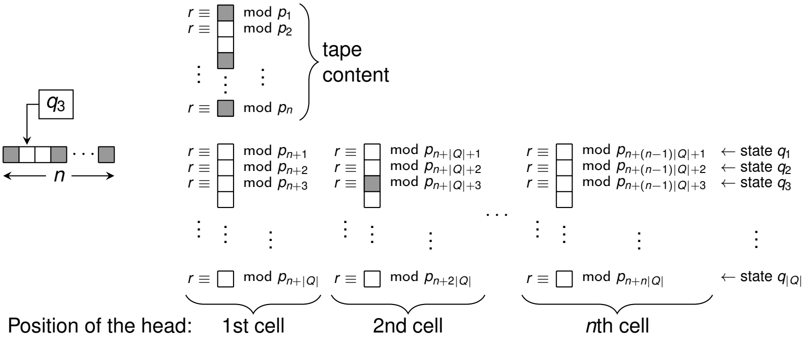
LBA to polynomial iteration

Let $p_1, \ldots, p_{n+n|Q|} \in$ PRIME.



LBA to polynomial iteration

Let $p_1, \ldots, p_{n+n|Q|} \in$ PRIME.



Simulating moves

To simulate a move of LBA from $[q_j, i, w]$ to $[q_k, i-1, w']$, where $w_i = 0$ and $w'_i = 1$, using a rule $\delta(q_j, 0) = (q_k, 1, L)$, we need to

Simulating moves

To simulate a move of LBA from $[q_j, i, w]$ to $[q_k, i-1, w']$, where $w_i = 0$ and $w'_i = 1$, using a rule $\delta(q_j, 0) = (q_k, 1, L)$, we need to

- verify that we are in the correct state q_j in position i;

$\delta(q_j, 0) = (q_k, 1, L)$

$\mathrm{EQONE}_{n+j+(i-1)|Q|}$

Simulating moves

To simulate a move of LBA from $[q_j, i, w]$ to $[q_k, i-1, w']$, where $w_i = 0$ and $w'_i = 1$, using a rule $\delta(q_j, 0) = (q_k, 1, L)$, we need to

- verify that we are in the correct state q_j in position i;
- move to state q_k in position $i-1$ from q_j in position i;

$$\delta(q_j, 0) = (q_k, 1, L) \quad \mathrm{FLIP}_{n+k+(i-2)|Q|} \circ \mathrm{FLIP}_{n+j+(i-1)|Q|} \circ \mathrm{EQONE}_{n+j+(i-1)|Q|}$$

Simulating moves

To simulate a move of LBA from $[q_j, i, w]$ to $[q_k, i-1, w']$, where $w_i = 0$ and $w'_i = 1$, using a rule $\delta(q_j, 0) = (q_k, 1, L)$, we need to

- verify that we are in the correct state q_j in position i;
- move to state q_k in position $i-1$ from q_j in position i;
- verify that the symbol in ith position is 0;

$$\delta(q_j, 0) = (q_k, 1, L)$$

0: EQZERO_i

q_j, q_k, L: $\mathrm{FLIP}_{n+k+(i-2)|Q|} \circ \mathrm{FLIP}_{n+j+(i-1)|Q|} \circ \mathrm{EQONE}_{n+j+(i-1)|Q|}$

Simulating moves

To simulate a move of LBA from $[q_j, i, w]$ to $[q_k, i-1, w']$, where $w_i = 0$ and $w'_i = 1$, using a rule $\delta(q_j, 0) = (q_k, 1, L)$, we need to

- verify that we are in the correct state q_j in position i;
- move to state q_k in position $i-1$ from q_j in position i;
- verify that the symbol in ith position is 0;
- rewrite that 0 as 1.

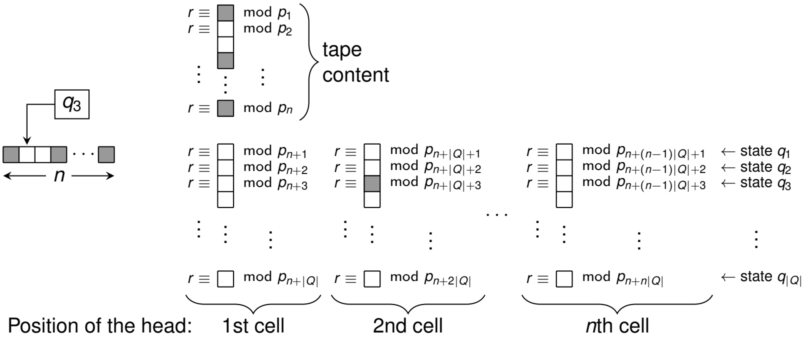
$$\delta(q_j, 0) = (q_k, 1, L)$$

$\mathrm{FLIP}_i \circ \mathrm{EQZERO}_i$

$\mathrm{FLIP}_{n+k+(i-2)|Q|} \circ \mathrm{FLIP}_{n+j+(i-1)|Q|} \circ \mathrm{EQONE}_{n+j+(i-1)|Q|}$

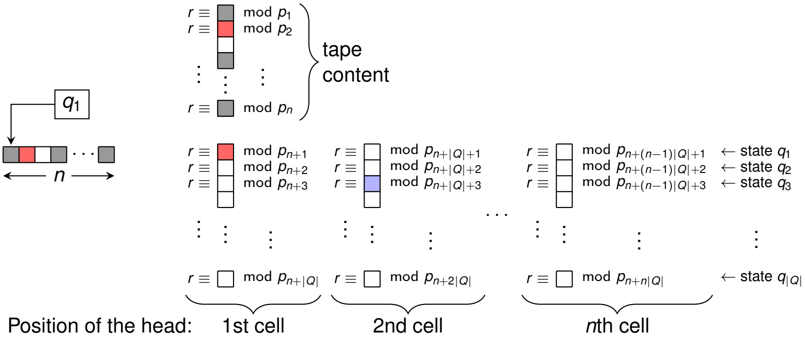
Simulating moves

Applying move $\delta(q_3, 0) = (q_1, 1, L)$ to $[q_3, 2, 1001 \cdots 1]$.



Simulating moves

Applying move $\delta(q_3, 0) = (q_1, 1, L)$ to $[q_3, 2, 1001 \cdots 1]$.



Final ingredients

- Initial integer x_0 satisfies

$$\begin{aligned} x_0 &\equiv 1 \mod p_{n+1} \\ x_0 &\equiv 0 \mod p_j. \end{aligned}$$

Final ingredients

- Initial integer x_0 satisfies

$$x_0 \equiv 1 \mod p_{n+1}$$
$$x_0 \equiv 0 \mod p_j.$$

- If LBA reaches $[q_f, 1, 0^n]$, then by simulating correctly

$$r \equiv 1 \mod p_{n+|Q|}$$
$$r \equiv 0 \mod p_j$$

can be reached. Then,

  - $p_{flip,n+|Q|}(p_{eqone,n+|Q|}(x))$ to reach $r \equiv 0 \mod p_i$ for all i.
  - $p(x) = x \pm p_1 \cdots p_{n+n|Q|}$ to reach the integer 0.

Final ingredients

- Initial integer x_0 satisfies

$$x_0 \equiv 1 \mod p_{n+1}$$
$$x_0 \equiv 0 \mod p_j.$$

- If LBA reaches $[q_f, 1, 0^n]$, then by simulating correctly

$$r \equiv 1 \mod p_{n+|Q|}$$
$$r \equiv 0 \mod p_j$$

  can be reached. Then,
  - $p_{flip,n+|Q|}(p_{eqone,n+|Q|}(x))$ to reach $r \equiv 0 \mod p_i$ for all i.
  - $p(x) = x \pm p_1 \cdots p_{n+n|Q|}$ to reach the integer 0.
- If LBA does not reach $[q_f, 1, 0^n]$, then simulating correctly will not result in 0.
  Simulating incorrectly results in $r \equiv 2 \mod p_i$ for some i.

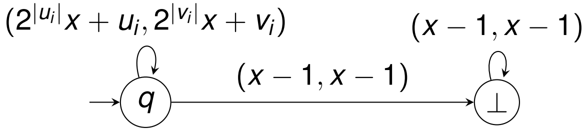
Higher dimensions

PRM in higher dimensions

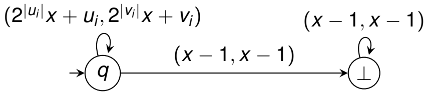
Theorem (Reichert 2015)

The reachability problem is undecidable for two-dimensional PRM, *where the updates are affine polynomials.*

Let $\{(u_1, v_1), \ldots, (u_n, v_n)\} \subseteq \{0,1\}^* \times \{0,1\}^*$ be an instance of the PCP.



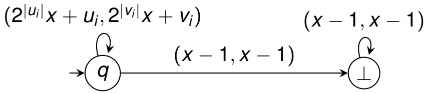
Polynomial iteration in higher dimensions



Let $p_1, p_2 \in$ PRIME. Consider polynomials

- $(2^{|u_i|}x + u_i, 2^{|v_i|}x + v_i, p_{eqone,1}(x))$ for each pair (u_i, v_i);
- $(x-1, x-1, p_{flip,2}(p_{flip,1}(p_{eqone,1}(x))))$;
- $(x-1, x-1, p_{eqone,2}(x))$;
- $(x, x, p_{flip,2}(p_{eqone,2}(x)))$ and $(x, x, x \pm p_1 p_2)$.

Polynomial iteration in higher dimensions



Let $p_1, p_2 \in$ PRIME. Consider polynomials

- $(2^{|u_i|}x + u_i, 2^{|v_i|}x + v_i, p_{eqone,1}(x))$ for each pair (u_i, v_i);
- $(x-1, x-1, p_{flip,2}(p_{flip,1}(p_{eqone,1}(x))))$;
- $(x-1, x-1, p_{eqone,2}(x))$;
- $(x, x, p_{flip,2}(p_{eqone,2}(x)))$ and $(x, x, x \pm p_1 p_2)$.

Theorem

The reachability problem for polynomial iteration is undecidable already for three-dimensional polynomials.

Conclusion

Summary

Theorem

Given $\mathcal{P} \subseteq \mathbb{Z}[x]$, the reachability problem for polynomial iteration is PSPACE*-complete.*

Model	Dimension		
	1	2	≥ 3
PRM	PSPACE-complete	U	–
stateless PRM	?	?	?

Summary

Theorem

Given $\mathcal{P} \subseteq \mathbb{Z}[x]$, the reachability problem for polynomial iteration is PSPACE*-complete.*

Model	Dimension		
	1	2	≥ 3
PRM	PSPACE-complete	U	–
stateless PRM	PSPACE-complete	?	U

Future work

- Decidability of two-dimensional polynomial iteration.
- Decidability of polynomial iteration over rational numbers in interval $[0, 1]$.
- Complexity of polynomial iteration over rational numbers.
- Investigate the effect of polynomials of the form $\pm x + b$ on the decidability of the reachability.

Thank you for your attention!