Undecidability for integer weighted Büchi automata and Robot Games with states

V. Halava[1] T. Harju[1] R. Niskanen[1] I. Potapov[2]

[1]Department of Mathematics and Statistics
University of Turku, Finland

[2]Department of Computer Science
University of Liverpool, UK

The Third Russian Finnish Symposium on Discrete Mathematics, 2014

Let $A = \{a_1, \ldots, a_j, \ldots, a_{m-1}\}$ be a finite alphabet.

- A^* consists of all finite words over A.
- A^ω consists of all infinite words over A.

Let $A = \{a_1, \ldots, a_j, \ldots, a_{m-1}\}$ be a finite alphabet and $w \in A^*$.

- A^* consists of all finite words over A.
- A^ω consists of all infinite words over A.
- $w(i)$ is ith letter of w.
- Let a be a letter. $aA^* = \{aw \mid w \in A^*\}$.

Let $A = \{a_1, \ldots, a_j, \ldots, a_{m-1}\}$ be a finite alphabet and $w \in A^*$.

- A^* consists of all finite words over A.
- A^ω consists of all infinite words over A.
- $w(i)$ is ith letter of w.
- Let a be a letter. $aA^* = \{aw \mid w \in A^*\}$.
- $u \leq w$ if $w = uv$ for some $v \in A^*$.
- $u < w$ if $w = uv$ for some non-empty $v \in A^*$

Let $A = \{a_1, \ldots, a_j, \ldots, a_{m-1}\}$ be a finite alphabet and $w \in A^*$.

- A^* consists of all finite words over A.
- A^ω consists of all infinite words over A.
- $w(i)$ is ith letter of w.
- Let a be a letter. $aA^* = \{aw \mid w \in A^*\}$.
- $u \leq w$ if $w = uv$ for some $v \in A^*$.
- $u < w$ if $w = uv$ for some non-empty $v \in A^*$
- Similarly defined when $w \in A^\omega$.

- For given Finite Automaton $\mathcal{A}$, over alphabet A, is its language $L(\mathcal{A}) = A^*$?
- For given Büchi Automaton $\mathcal{B}$, over alphabet A, is its language $L(\mathcal{B}) = A^\omega$?

- For given Finite Automaton $\mathcal{A}$, over alphabet A, is its language $L(\mathcal{A}) = A^*$?
- For given Büchi Automaton $\mathcal{B}$, over alphabet A, is its language $L(\mathcal{B}) = A^\omega$?
- Both are known to be decidable.

- Extend automaton by adding *weight function* γ to transitions.
- For given Weighted Automaton $\mathcal{A}^\gamma$, over alphabet *A*, is its language $L(\mathcal{A}^\gamma) = A^*$?

- Extend automaton by adding *weight function* γ to transitions.
- For given Weighted Automaton $\mathcal{A}^{\gamma}$, over alphabet *A*, is its language $L(\mathcal{A}^{\gamma}) = A^*$?
- Shown to be undecidable by Halava and Harju in 1999.

- Let $\mathcal{A} = (Q, A, \delta, q_0)$ be a finite automaton, where Q is set of states, A is alphabet, δ is transition function, and q_0 is initial state.

- Let $\mathcal{A} = (Q, A, \delta, q_0)$ be a finite automaton, where Q is set of states, A is alphabet, δ is transition function, and q_0 is initial state.
- Instead of δ, consider σ, which is a mapping of edges T to transitions.
- Let $\gamma : T \to \mathbb{Z}$ be the weight function that assigns weights to edges.

- Let $\mathcal{A} = (Q, A, \delta, q_0)$ be a finite automaton, where Q is set of states, A is alphabet, δ is transition function, and q_0 is initial state.
- Instead of δ, consider σ, which is a mapping of edges T to transitions.
- Let $\gamma : T \to \mathbb{Z}$ be the weight function that assigns weights to edges.
- Let w be an infinite word. It is accepted by $\mathcal{A}^\gamma$ if there exists a computation π such that infinite number of its prefixes p have zero weight.

- We use undecidability of *Infinite Post Correspondence Problem* (ωPCP) for our proof.

- We use undecidability of *Infinite Post Correspondence Problem* (ωPCP) for our proof.
- In ωPCP we are given two morphisms $h, g : A^* \to B^*$.
- Does there exist an infinite word w such that for all prefixes p either $h(p) < g(p)$ or $g(p) < h(p)$?

- We use undecidability of *Infinite Post Correspondence Problem* (ωPCP) for our proof.
- In ωPCP we are given two morphisms $h, g : A^* \to B^*$.
- Does there exist an infinite word w such that for all prefixes p either $h(p) < g(p)$ or $g(p) < h(p)$?
- Shown to be undecidable by Halava and Harju for domain alphabets $|A| \geq 9$ and improved to $|A| \geq 8$ by Dong and Liu.

- The idea of the proof of ωPCP is to simulate a semi-Thue system with undecidable termination problem.

- The idea of the proof of ωPCP is to simulate a semi-Thue system with undecidable termination problem.
- The solutions are of specific form:
  1. The first letter is d that contains the input of semi-Thue system.
  2. An infinite sequence of configurations separated by special symbols $\#$.

- The idea of the proof of ωPCP is to simulate a semi-Thue system with undecidable termination problem.
- The solutions are of specific form:
  1. The first letter is d that contains the input of semi-Thue system.
  2. An infinite sequence of configurations separated by special symbols $\#$.
  3. Let p be a prefix of a solution, then $|g(p)| < |h(p)|$.
  4. "Image under h is ahead of image under g by one configuration".

- The goal is to construct an automaton $\mathcal{A}^\gamma$ such that $L(\mathcal{A}^\gamma) = A^\omega$ if and only if the instance of ωPCP has no solution.

- The goal is to construct an automaton $\mathcal{A}^\gamma$ such that $L(\mathcal{A}^\gamma) = A^\omega$ if and only if the instance of ωPCP has no solution.
- An infinite word $w \in dA^\omega$ is accepted by $\mathcal{A}^\gamma$ if and only if for some finite prefix p of w, $g(p) \not< h(p)$. Such a prefix p does exist for all infinite words except for the solutions of the instance (h, g). We call the verification of such a prefix p *error checking*.
- All infinite words beginning with letter in $A \setminus \{d\}$ are accepted by $\mathcal{A}^\gamma$.

First for each $a \in A$, let

$$\langle q_0, a, q_0, m(|h(a)| - |g(a)|)\rangle\,, \quad \langle q_1, a, q_1, m(-|g(a)|)\rangle\,, \quad \langle q_2, a, q_2, 0\rangle$$

be in σ, and for all $b \in A \setminus \{d\}$, let

$$\langle q_0, b, q_2, 0\rangle \in \sigma.$$

For error checking we need the following transitions for all letters $a \in A$: Let $h(a) = b_{j_1} b_{j_2} \cdots b_{j_{n_1}}$, where $b_{j_i} \in B$, for each index $1 \leq k \leq n_1$. Then let, for each $k = 1, \ldots, n_1$,

$$\langle q_0, a, q_1, m(k - |g(a)|) + j_k\rangle \in \sigma \tag{1}$$

Let $g(a) = b_{i_1} b_{i_2} \cdots b_{i_{n_2}}$, where $b_{i_\ell} \in B$, for each index $1 \leq \ell \leq n_2$. For each $\ell = 1, \ldots, n_2$ and letter $b_c \in B$ such that $b_{i_\ell} \neq b_c \in B$, let

$$\langle q_1, a, q_2, -m\ell - c\rangle \in \sigma. \tag{2}$$

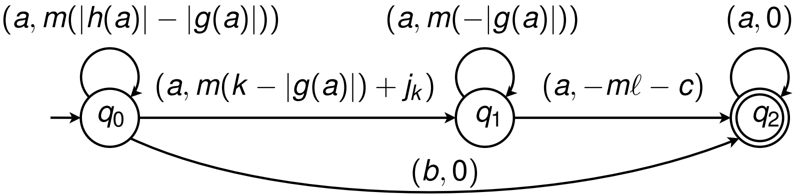
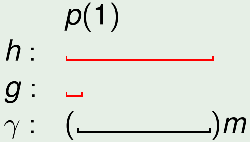
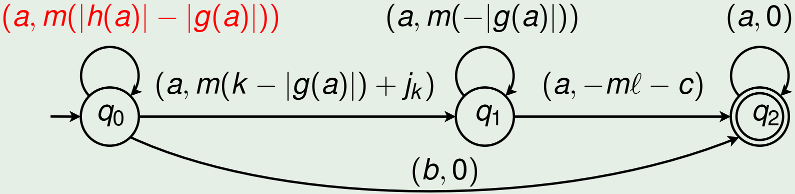


Figure: The weighted automaton $\mathcal{A}^\gamma$. In the figure $a \in A, b \in A \setminus \{d\}$ and $|A| = m - 1$.

Example

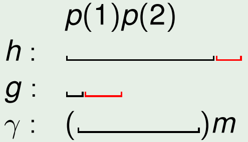
Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

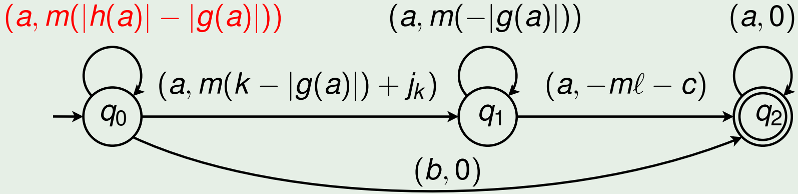




Example

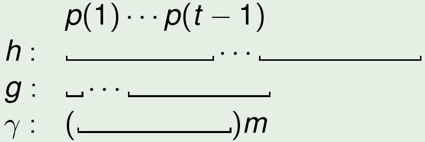
Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

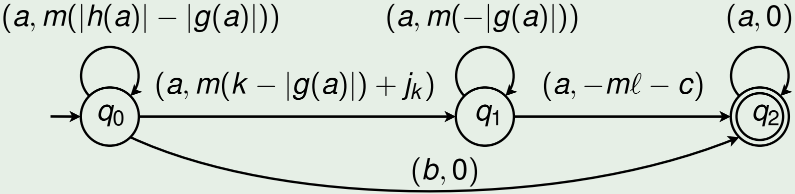




Example

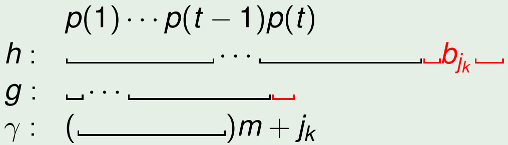
Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

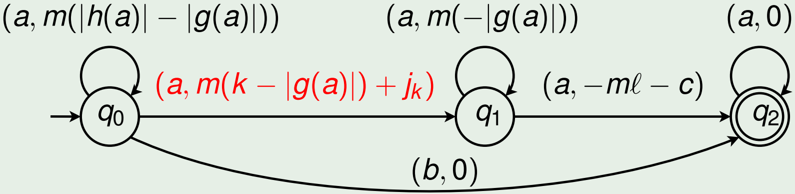




Example

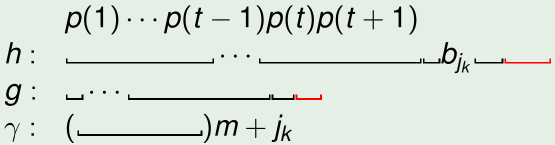
Let *dw* be an infinite word that is not a solution for ωPCP and let *p* be its prefix such that for some index *r* $h(p)(r) \neq g(p)(r)$.

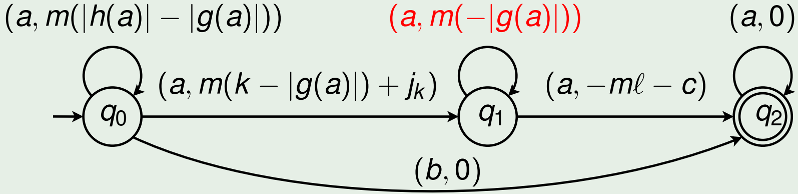




Example

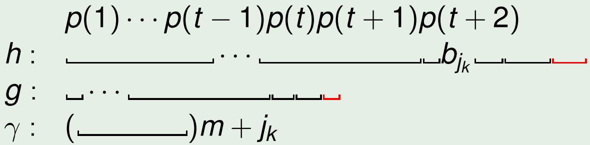
Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

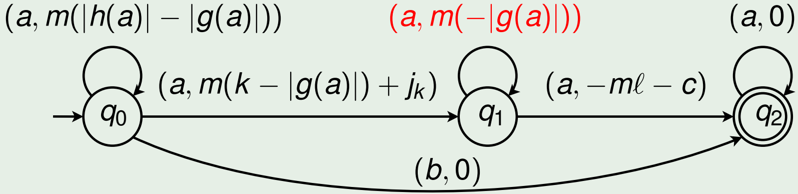




Example

Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.





Example

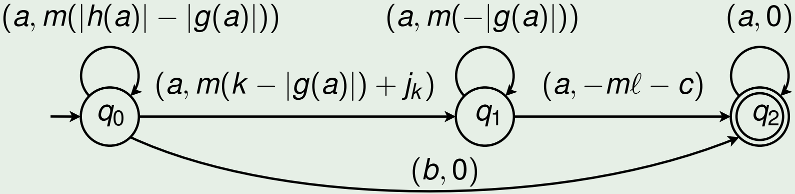
Let *dw* be an infinite word that is not a solution for ωPCP and let *p* be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

$p(1)\cdots p(t-1)p(t)p(t+1)\cdots p(n-1)$

h : ␣ $\cdots$ ␣ b_{j_k} ␣ ␣ $\cdots$ ␣

g : ␣ $\cdots$ ␣ ␣ ␣ $\cdots$ ␣

γ : $(␣)m + j_k$



Example

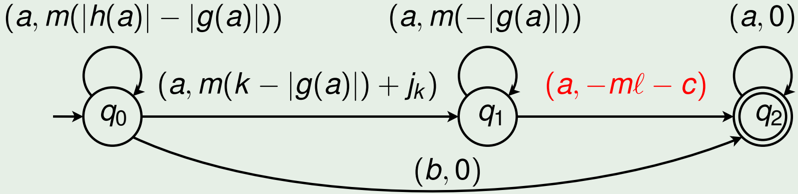
Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

$$p(1)\cdots p(t-1)p(t)p(t+1)\cdots p(n-1)p(n)$$

h : $\sqcup \cdots \sqcup b_{j_k} \sqcup \cdots \sqcup$

g : $\sqcup \cdots \sqcup \cdots \sqcup b_\ell \sqcup$

γ : 0



Example

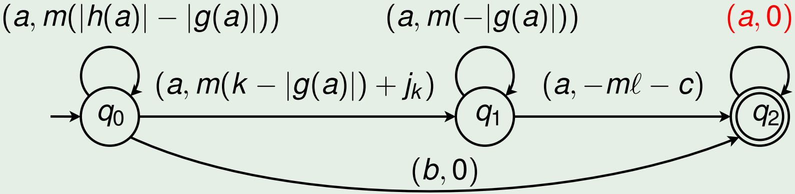
Let dw be an infinite word that is not a solution for ωPCP and let p be its prefix such that for some index r $h(p)(r) \neq g(p)(r)$.

$$p(1)\cdots p(t-1)p(t)p(t+1)\cdots p(n-1)p(n)p(n+1)$$

h : $\cdots$ b_{j_k} $\cdots$

g : $\cdots$ $\cdots$ b_ℓ

γ : 0



Theorem

It is undecidable whether or not $L(\mathcal{A}^\gamma) = A^\omega$ holds for 3-state integer weighted Büchi automata $\mathcal{A}^\gamma$ over its alphabet A.

Theorem

It is undecidable whether or not L$(\mathcal{A}^\gamma) = A^\omega$ holds for 3-state integer weighted Büchi automata $\mathcal{A}^\gamma$ over its alphabet A.

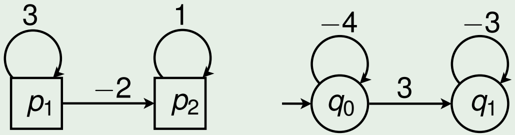
Corollary

It is undecidable whether or not L$(\mathcal{B}^\gamma) = A^\omega$ holds for 3-state integer weighted Büchi automata $\mathcal{B}^\gamma$, where each state is final, over its alphabet A.

- Robot Game with states is a two-player vector addition game on integer lattice $\mathbb{Z}^n$.
- From initial point $\mathbf{x}_0$, *Attacker* tries to reach the origin, while *Defender* tries to keep the game from reaching the origin.

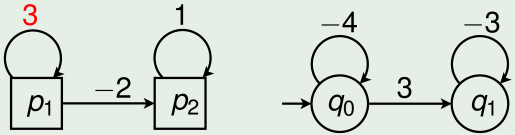
- Robot Game with states is a two-player vector addition game on integer lattice $\mathbb{Z}^n$.
- From initial point $\mathbf{x}_0$, *Attacker* tries to reach the origin, while *Defender* tries to keep the game from reaching the origin.
- Vectors players can play are dependant on the state of the underlying automaton.
- The decision problem is whether there exists a winning strategy for Attacker. That is, can Attacker win, regardless of the vectors Defender plays.

Example



Consider one-dimensional game with initial point $x_0 = 2$.

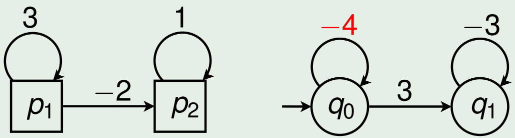
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 5$

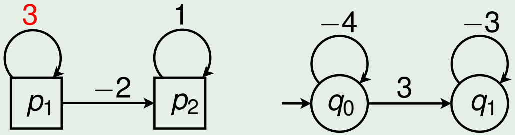
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \rightarrow 5 \rightarrow 1$

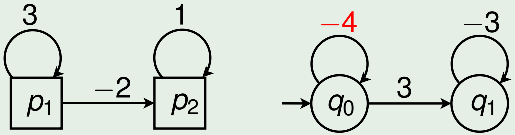
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 5 \to 1 \to 4$

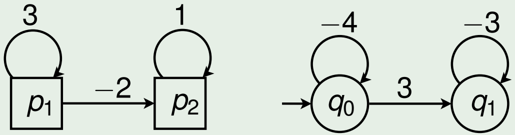
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$$2 \to 5 \to 1 \to 4 \to 0$$

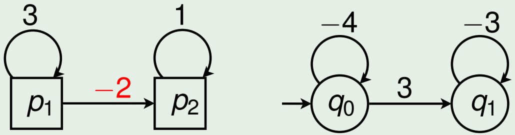
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \rightarrow 5 \rightarrow 1$

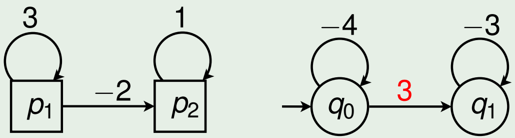
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \rightarrow 5 \rightarrow 1 \rightarrow -1$

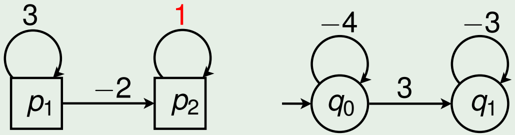
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 5 \to 1 \to -1 \to 2$

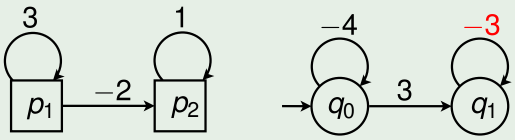
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 5 \to 1 \to -1 \to 2 \to 3$

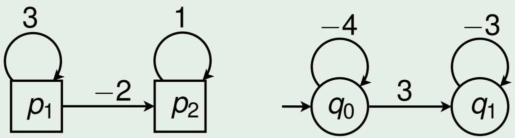
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$$2 \to 5 \to 1 \to -1 \to 2 \to 3 \to 0$$

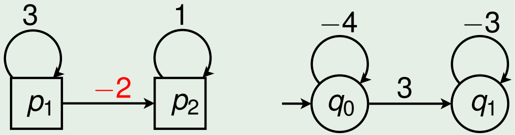
Example



Consider one-dimensional game with initial point $x_0 = 2$.

2

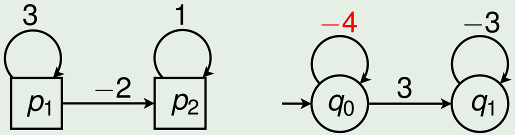
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 0$

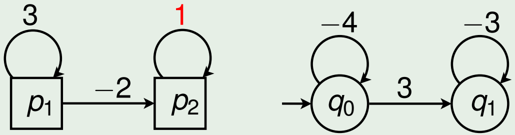
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 0 \to -4$

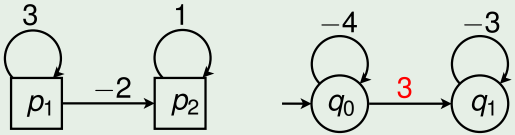
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$2 \to 0 \to -4 \to -3$

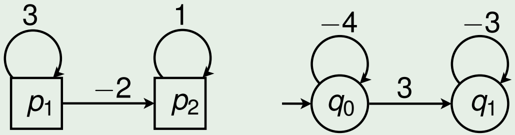
Example



Consider one-dimensional game with initial point $x_0 = 2$.

$$2 \to 0 \to -4 \to -3 \to 0$$

Example



Consider one-dimensional game with initial point $x_0 = 2$.

$$2 \to 0 \to -4 \to -3 \to 0$$

$\therefore$ Attacker has a winning strategy.

- We construct a Robot Game with states where Attacker has a winning strategy if and only if the underlying automaton is universal.

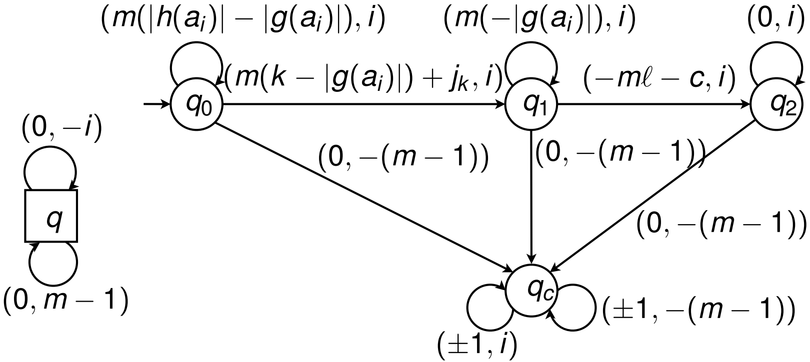


Figure: The picture of the weighted automata $\mathcal{B}$ and $\mathcal{A}$ with vectors substituting the transitions. In the figure $a_i \in A$.

- The idea is that Defender gives letters in second component of the vector and Attacker has to match the letter as well as modify first component according to the automaton.

- The idea is that Defender gives letters in second component of the vector and Attacker has to match the letter as well as modify first component according to the automaton.
- If Attacker does not match the letter, Defender wins using the special move.

- The idea is that Defender gives letters in second component of the vector and Attacker has to match the letter as well as modify first component according to the automaton.
- If Attacker does not match the letter, Defender wins using the special move.
- If Defender abuses the special move, Attacker wins by going to new checking state.

- The idea is that Defender gives letters in second component of the vector and Attacker has to match the letter as well as modify first component according to the automaton.
- If Attacker does not match the letter, Defender wins using the special move.
- If Defender abuses the special move, Attacker wins by going to new checking state.
- If Attacker abuses the checking state, Defender wins.

- The idea is that Defender gives letters in second component of the vector and Attacker has to match the letter as well as modify first component according to the automaton.
- If Attacker does not match the letter, Defender wins using the special move.
- If Defender abuses the special move, Attacker wins by going to new checking state.
- If Attacker abuses the checking state, Defender wins.
- If the word played by Defender is not accepted by the automaton, the first component will never be 0.

- The idea is that Defender gives letters in second component of the vector and Attacker has to match the letter as well as modify first component according to the automaton.
- If Attacker does not match the letter, Defender wins using the special move.
- If Defender abuses the special move, Attacker wins by going to new checking state.
- If Attacker abuses the checking state, Defender wins.
- If the word played by Defender is a solution of ωPCP, first component will never be 0.

Theorem

It is undecidable whether Attacker has a winning strategy in 2-dimensional Robot Game with states.

Theorem

It is undecidable whether Attacker has a winning strategy in 2-dimensional Robot Game with states.

Note that, while number of states is small for both players, for Defender number of edges is also small but for Attacker there are millions of edges.

- Universality Problem for 2-state integer weighted Büchi automata.
- An upper bound on number of edges in Robot Games with states.
- Application to similar Defender-Attacker games.

THANK YOU!