Weighted Automata on Infinite Words in the Context of Attacker-Defender Games

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

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

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

Computability in Europe, 2015

Attacker-Defender Games

- Two players: Attacker, Defender.
- Players play in turns using available moves.
- Initial and target configurations.
- Configuration is a sequence of alternating moves.
- Play is an infinite sequence of configurations.
- Attacker wins if the target configuration is reachable in a play starting from the initial configuration. Otherwise Defender wins.

Games we consider

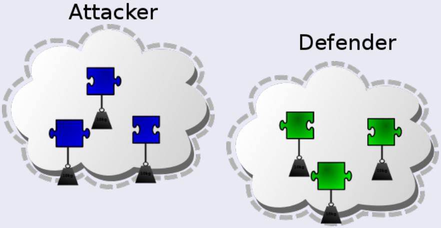
1. Weighted Word Games
2. Word Game on pairs of group words
3. Matrix Games on vectors
4. Braid Games

Theorem

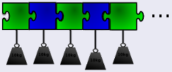
It is undecidable whether Attacker has a winning strategy in these games.

Weighted Word Games

- Players are given sets of words over free group alphabet.



- They play words in turns.

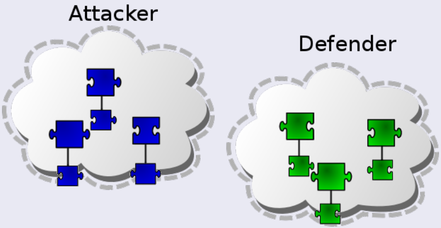
concatenation   $a\bar{b}ab$

sum -7

- Attacker's goal to reach a certain word with zero weight.

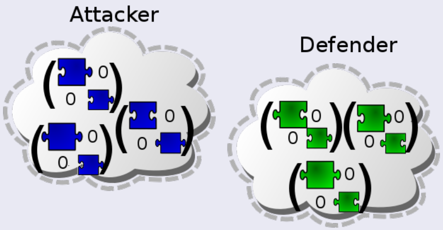
Word Games on pairs of group words

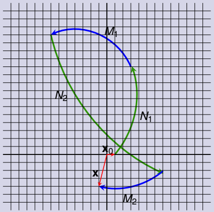
- Similar to Weighted Word Games but now the weight is encoded as a unary word.



Matrix Games

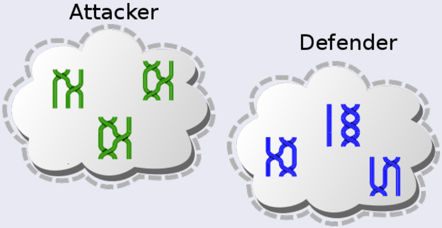
- Players are given sets of matrices from $SL(n, \mathbb{Z})$.
  - Attacker: $\{M_1, \ldots, M_k\}$
  - Defender: $\{N_1, \ldots, N_\ell\}$
  - Initial and target vectors
- We encode words from Word Games on pair of words into 4×4 matrices.

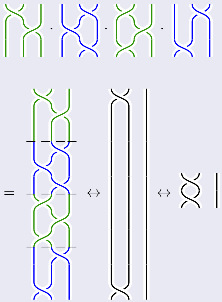




Braid Games

- Two variants - played on 3 (B_3) or 5 strands (B_5).
- Players are given sets of braid words.
- Target is a braid isotopic to a trivial braid.

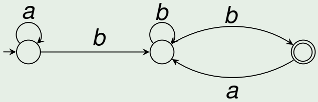




Universality Problem for Finite Automata

- 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.

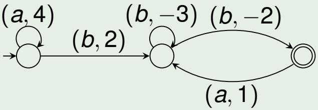
Example



Universality Problem for Weighted Automata

- 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.

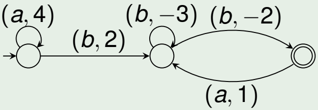
Example



Universality Problem for Weighted Automata

- 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.

Example



Unfortunately this result is not strong enough for games.

Weighted Automaton on Infinite Words

Let $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ be a finite automaton, where Q is the set of states, A is the alphabet, σ is the set of transitions, q_0 is the initial state, F is the set of final states, and $\mathbb{Z}$ is the additive group of integers.

Weighted Automaton on Infinite Words

Let $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ be a finite automaton, where Q is the set of states, A is the alphabet, σ is the set of transitions, q_0 is the initial state, F is the set of final states, and $\mathbb{Z}$ is the additive group of integers.

Transitions

In the form
$t = \langle q, a, q', z \rangle$.

Weighted Automaton on Infinite Words

Let $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ be a finite automaton, where Q is the set of states, A is the alphabet, σ is the set of transitions, q_0 is the initial state, F is the set of final states, and $\mathbb{Z}$ is the additive group of integers.

Transitions

In the form $t = \langle q, a, q', z \rangle$.

Weight of a path

- Let $\pi = t_{i_0} t_{i_1} \cdots$ be an infinite path of $\mathcal{A}$. Let $p \leq \pi$ be a finite prefix.
- Weight of p is $\gamma(p) = z_0 + \ldots + z_n \in \mathbb{Z}$.

Weighted Automaton on Infinite Words

Let $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ be a finite automaton, where Q is the set of states, A is the alphabet, σ is the set of transitions, q_0 is the initial state, F is the set of final states, and $\mathbb{Z}$ is the additive group of integers.

Transitions

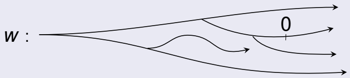
In the form $t = \langle q, a, q', z \rangle$.

Weight of a path

- Let $\pi = t_{i_0} t_{i_1} \cdots$ be an infinite path of $\mathcal{A}$. Let $p \leq \pi$ be a finite prefix.
- Weight of p is $\gamma(p) = z_0 + \ldots + z_n \in \mathbb{Z}$.

Acceptance condition

Let $w \in A^\omega$. It is accepted by $\mathcal{A}$ if there exists a computation π such that for some prefix p has zero weight.



Weighted Automaton on Infinite Words

Let $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ be a finite automaton, where Q is the set of states, A is the alphabet, σ is the set of transitions, q_0 is the initial state, F is the set of final states, and $\mathbb{Z}$ is the additive group of integers.

Transitions

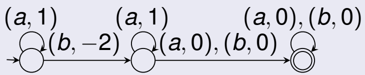
In the form $t = \langle q, a, q', z \rangle$.

Weight of a path

- Let $\pi = t_{i_0} t_{i_1} \cdots$ be an infinite path of $\mathcal{A}$. Let $p \leq \pi$ be a finite prefix.
- Weight of p is $\gamma(p) = z_0 + \ldots + z_n \in \mathbb{Z}$.

Acceptance condition

Let $w \in A^\omega$. It is accepted by $\mathcal{A}$ if there exists a computation π such that for some prefix p has zero weight.



Weighted Automaton on Infinite Words

Let $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ be a finite automaton, where Q is the set of states, A is the alphabet, σ is the set of transitions, q_0 is the initial state, F is the set of final states, and $\mathbb{Z}$ is the additive group of integers.

Transitions

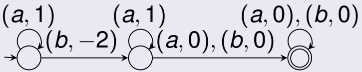
In the form $t = \langle q, a, q', z \rangle$.

Weight of a path

- Let $\pi = t_{i_0} t_{i_1} \cdots$ be an infinite path of $\mathcal{A}$. Let $p \leq \pi$ be a finite prefix.
- Weight of p is $\gamma(p) = z_0 + \ldots + z_n \in \mathbb{Z}$.

Acceptance condition

Let $w \in A^\omega$. It is accepted by $\mathcal{A}$ if there exists a computation π such that for some prefix p has zero weight.



Words beginning with *aab*, *aba* or *baa* are accepted.

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]$$

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]\left[\frac{bb}{baa}\right]$$

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]\left[\frac{bb}{baa}\right]\left[\frac{aaa}{aa}\right]$$

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]\left[\frac{bb}{baa}\right]\left[\frac{aaa}{aa}\right]\left[\frac{aaa}{aa}\right]$$

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]\left[\frac{bb}{baa}\right]\left[\frac{aaa}{aa}\right]\left[\frac{aaa}{aa}\right]$$

is a solution since both halves read *abbbaaaaaa*.

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]\left[\frac{bb}{baa}\right]\left[\frac{aaa}{aa}\right]\left[\frac{ab}{abb}\right]$$

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Example

Consider $P = \left\{ \left[\frac{ab}{abb}\right], \left[\frac{bb}{baa}\right], \left[\frac{aaa}{aa}\right] \right\}$.

$$\left[\frac{ab}{abb}\right]\left[\frac{bb}{baa}\right]\left[\frac{aaa}{aa}\right]\left[\frac{ab}{abb}\right]$$

is not a solution as $abbbaaaab \neq abbbaaaabb$

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Theorem (Matiyasevich, Senizergues 05)

It is undecidable whether PCP with 7 dominoes has a solution.

Post Correspondence Problem (Post 46)

Given a finite set of dominoes with words on top and bottom halves, can we construct a finite sequence of dominoes where words on top and bottom halves are equal.

Theorem (Matiyasevich, Senizergues 05)

It is undecidable whether PCP with 7 dominoes has a solution.

Proof (idea)

Turing Machine can be simulated with dominoes.

$$\left[\frac{}{conf_0}\right]\left[\frac{conf_0 \cdot conf_1 \cdots}{conf_1 \cdot conf_2 \cdots}\right]\left[\frac{conf_{halt}}{}\right]$$

Constructed in such way that words are equal if and only if TM halts.

Infinite Post Correspondence Problem (ωPCP)

- 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.

Example

Consider $P = \left\{ \left[\frac{ab}{abb} \right], \left[\frac{bb}{baa} \right], \left[\frac{aaa}{aa} \right] \right\}$. It has an infinite solution

$$\left[\frac{aaa}{aa} \right] \left[\frac{aaa}{aa} \right] \left[\frac{aaa}{aa} \right] \cdots$$

Application of PCP

- Typically dominoes and building a sequence of dominoes are encoded into the model.
- The whole computation is stored.
- For this addition dimensions are required.

Application of PCP

- Typically dominoes and building a sequence of dominoes are encoded into the model.
- The whole computation is stored.
- For this addition dimensions are required.

Our technique

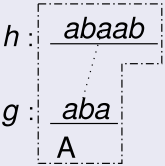
- ω PCP is not modeled by the automaton!

1. We guess the position where letters will be unequal.
2. Then we verify that this indeed happens.
3. Can be done with only one counter.

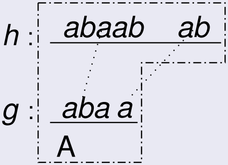
Idea of construction

- The goal is to construct an automaton $\mathcal{A}$ such that $L(\mathcal{A}) = A^\omega$ if and only if the instance of ωPCP has no solution.
- An infinite word $w \in A^\omega$ is accepted by $\mathcal{A}$ if and only if for some finite prefix p of w, $g(p) \not< h(p)$ and $h(p) \not< g(p)$.
- 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*.

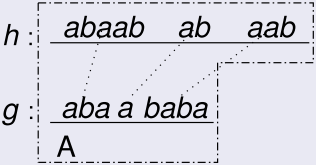
One possible path in the automaton



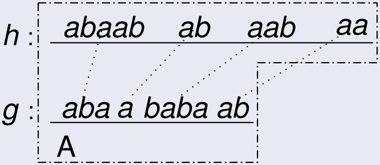
One possible path in the automaton



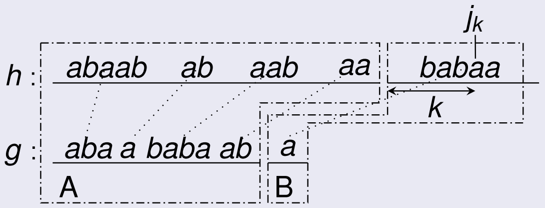
One possible path in the automaton



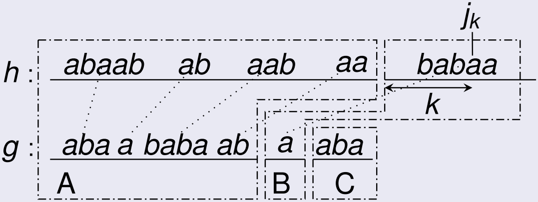
One possible path in the automaton



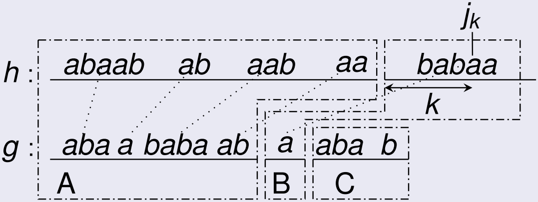
One possible path in the automaton



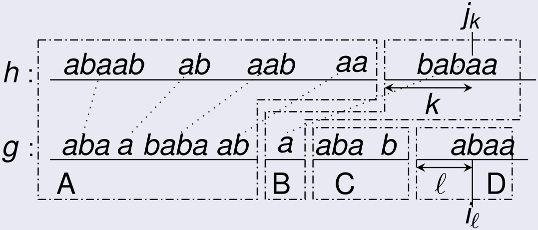
One possible path in the automaton



One possible path in the automaton



One possible path in the automaton

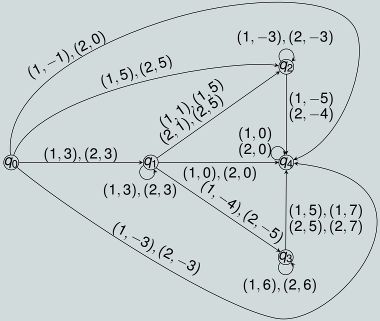


Example

ω PCP instance

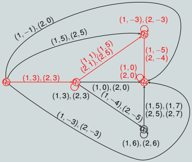
Let $g(1) = ab, g(2) = ab$, $h(1) = a, h(2) = b$.

Corresponding automaton



Example

Corresponding automaton



ω PCP instance

Let $g(1) = ab, g(2) = ab,$
$h(1) = a, h(2) = b.$

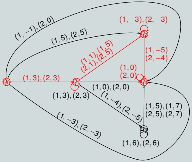
$w = 12122\cdots$ is accepted.

Example

ω PCP instance

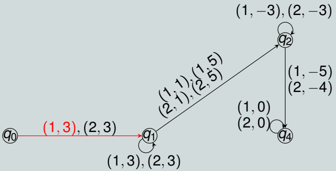
Let $g(1) = ab, g(2) = ab,$
$h(1) = a, h(2) = b.$

Corresponding automaton



$w = 12122\cdots$ is accepted.

Relevant part of the automaton



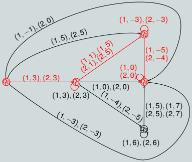
$p = 1$
$g(p) = ab$
$h(p) = a$
$\gamma(p) = 3$

Example

ω PCP instance

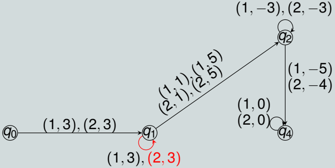
Let $g(1) = ab, g(2) = ab$, $h(1) = a, h(2) = b$.

Corresponding automaton



$w = 12122\cdots$ is accepted.

Relevant part of the automaton



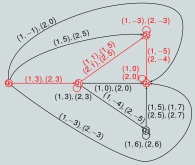
$p = 12$
$g(p) = abab$
$h(p) = ab$
$\gamma(p) = 3 + 3 = 6$

Example

ω PCP instance

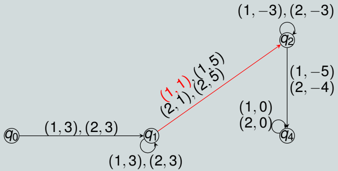
Let $g(1) = ab, g(2) = ab$, $h(1) = a, h(2) = b$.

Corresponding automaton



$w = 12122\cdots$ is accepted.

Relevant part of the automaton



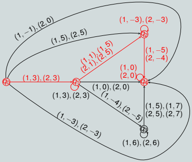
$p = 121$
$g(p) = ababab$
$h(p) = aba$
$\gamma(p) = 3 + 3 + 1 = 7$

Example

ω PCP instance

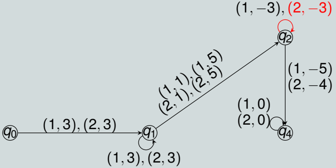
Let $g(1) = ab, g(2) = ab$, $h(1) = a, h(2) = b$.

Corresponding automaton



$w = 12122\cdots$ is accepted.

Relevant part of the automaton



$p = 1212$

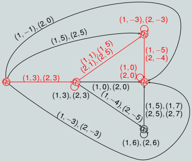
$g(p) = abababab$

$h(p) = abab$

$\gamma(p) = 3 + 3 + 1 + (-3) = 4$

Example

Corresponding automaton

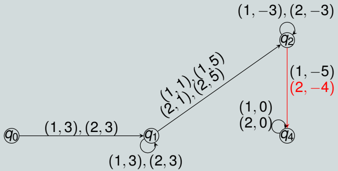


ω PCP instance

Let $g(1) = ab, g(2) = ab,$
$h(1) = a, h(2) = b.$

$w = 12122\cdots$ is accepted.

Relevant part of the automaton



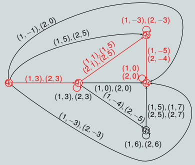
$p = 12122$
$g(p) = ababababab$
$h(p) = ababb$
$\gamma(p) = 3 + 3 + 1 - 3 + (-4) = 0$

Example

ω PCP instance

Let $g(1) = ab, g(2) = ab,$
$h(1) = a, h(2) = b.$

Corresponding automaton



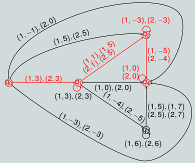
$w = (12)^\omega$ is not accepted.

Example

ω PCP instance

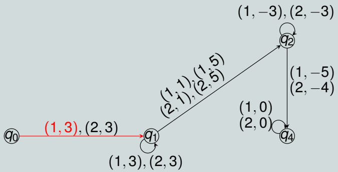
Let $g(1) = ab, g(2) = ab,$
$h(1) = a, h(2) = b.$

Corresponding automaton



$w = (12)^\omega$ is not accepted.

Relevant part of the automaton



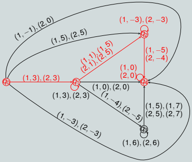
$p = 1$
$g(p) = ab$
$h(p) = a$
$\gamma(p) = 3$

Example

ω PCP instance

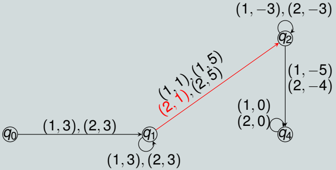
Let $g(1) = ab, g(2) = ab$,
$h(1) = a, h(2) = b$.

Corresponding automaton



$w = (12)^\omega$ is not accepted.

Relevant part of the automaton



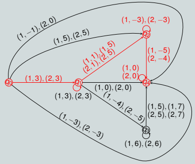
$p = 12$
$g(p) = abab$
$h(p) = ab$
$\gamma(p) = 3 + 1 = 4$

Example

ω PCP instance

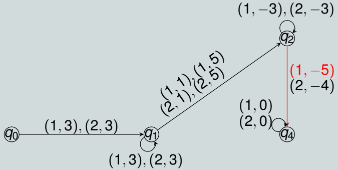
Let $g(1) = ab, g(2) = ab,$
$h(1) = a, h(2) = b.$

Corresponding automaton



$w = (12)^\omega$ is not accepted.

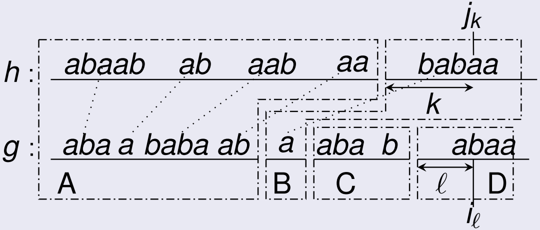
Relevant part of the automaton



$p = 121$
$g(p) = ababab$
$h(p) = aba$
$\gamma(p) = 3 + 1 + (-5) = -1$

Additional paths are needed for different forms of PCP instances:

- Image under *h* is longer and error is far away.
  - (part C is non-empty)
- Image under *g* is longer and error is far away.
  - (part C is non-empty)
- Image under *h* is longer and error is close.
  - (part C is empty, parts B and D are done simultaneously)
- Image under *g* is longer and error is close.
  - (part C is empty, parts B and D are done simultaneously)



Theorem

It is undecidable whether or not $L(\mathcal{A}) = A^\omega$ holds for 5-state integer weighted automata $\mathcal{A}$ on infinite words over alphabet A.

Attacker-Defender Games

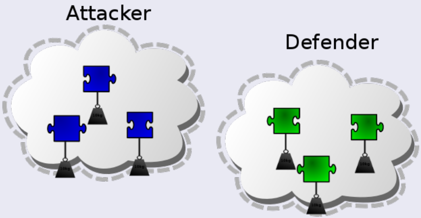
- Two players: Attacker, Defender.
- Players play in turns using available moves.
- Initial and target configurations.
- Configuration is a sequence of alternating moves.
- Play is an infinite sequence of configurations.
- Attacker wins if target configuration is in reachable in a play starting from initial configuration. Otherwise Defender wins.

Games we consider

1. Weighted Word Games
2. Word Game on pairs of group words
3. Matrix Games on vectors
4. Braid Games

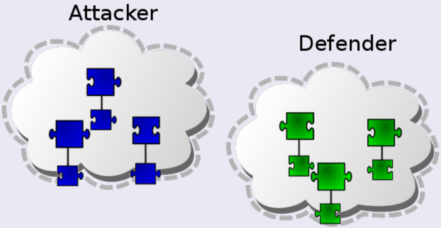
Weighted Word Games

- Players are given sets of words over free group alphabet.
- Defender plays a word of the automaton letter by letter.
- Attacker plays words corresponding to the letter played by Defender and respecting the structure of the automaton.
- Attacker tries to reach the word corresponding to the accepting configuration starting from the word corresponding to the initial configuration.
- Attacker has a winning strategy if and only if every word that Defender plays is accepted, that is the automaton is universal.



Word Games on pairs of group words

- Similar to Weighted Word Games but now the weight is encoded as a unary word.
- Using an additional trick, initial and final words are $(\varepsilon, \varepsilon)$.
- Attacker has a winning strategy if and only if every word that Defender plays is accepted, that is the automaton is universal.



Idea of construction for Word Game

- Initial word q_0.
- Defender plays letter a.
  Current word: $q_0 a$

Idea of construction for Word Game

- Initial word q_0.
- Defender plays letter a.
  Current word: $q_0 a$
- Attacker plays word $\overline{a}\,\overline{q_0}q_1$ corresponding to transition $\langle q_0, a, q_1 \rangle$.
  Current word: $q_0 a(\overline{a}\,\overline{q_0}q_1) = q_1$

Idea of construction for Word Game

- Initial word q_0.
- Defender plays letter a.
  Current word: $q_0 a$
- Attacker plays word $\overline{a}\,\overline{q_0} q_1$ corresponding to transition $\langle q_0, a, q_1 \rangle$.
  Current word: $q_0 a(\overline{a}\,\overline{q_0} q_1) = q_1$
- Defender plays letter b.
  Current word: $q_1 b$

Idea of construction for Word Game

- Initial word q_0.
- Defender plays letter a.
  Current word: $q_0 a$
- Attacker plays word $\overline{a}\,\overline{q_0} q_1$ corresponding to transition $\langle q_0, a, q_1 \rangle$.
  Current word: $q_0 a(\overline{a}\,\overline{q_0} q_1) = q_1$
- Defender plays letter b.
  Current word: $q_1 b$
- Attacker plays word $\overline{b}\overline{q_1} q_2$ corresponding to transition $\langle q_1, b, q_2 \rangle$.
  Current word: $q_1 b(\overline{b}\overline{q_1} q_2) = q_2$

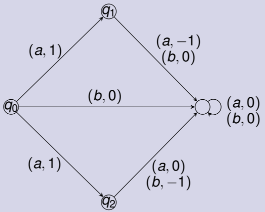
Idea of construction for Word Game

- Initial word q_0.
- Defender plays letter a.
  Current word: $q_0 a$
- Attacker plays word $\overline{a}\,\overline{q_0}q_1$ corresponding to transition $\langle q_0, a, q_1\rangle$.
  Current word: $q_0 a(\overline{a}\,\overline{q_0}q_1) = q_1$
- Defender plays letter b.
  Current word: $q_1 b$
- Attacker plays word $\overline{b}\overline{q_1}q_2$ corresponding to transition $\langle q_1, b, q_2\rangle$.
  Current word: $q_1 b(\overline{b}\overline{q_1}q_2) = q_2$
- If Attacker does not match the letter or plays incorrect transition, uncancellable elements will remain.
- In actual Word Game, the construction is slightly more complicated.

Counter Example

(In previous construction, Attacker commits to a path, while Defender does not commit to a word.)

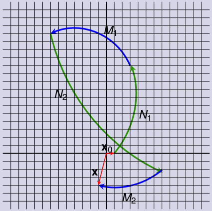
Universal automaton



- Now from q_0, Defender plays a.
- If Attacker plays $\overline{a}\,\overline{q_0}q_1$, then Defender will play only b and there is no path with 0 weight.
- If Attacker plays $\overline{a}\,\overline{q_0}q_2$, then Defender will play only a and there is no path with 0 weight.

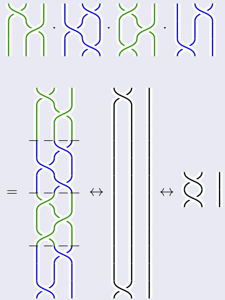
Matrix Games

- Players are given sets of matrices from $SL(n, \mathbb{Z})$.
- Starting from initial vector $\mathbf{x}_0$, players apply their matrices in turns.
- Attacker's goal is to reach $\mathbf{x}_0$.
- We encode words from Word Games on pair of words into 4×4 matrices.
- Since matrices are from $SL(4, \mathbb{Z})$, this is only possible when matrix played by the players is the identity matrix.
- Identity matrix is reachable if and only if the empty word is reachable in the Word Game.



Braid Games

- Two variants - played on 3 (B_3) or 5 strands (B_5).
- Players are given sets of braid words.
- In B_3, starting from a braid word corresponding to the initial word of Weighted Word Game, Attacker's aim is to reach the trivial braid.
- B_5 contains direct product of two free groups of rank 2 as a subgroup.
- We encode words of Word Game on pair of words into braids.
- In both variants, the trivial braid is reachable if and only if the empty word is reachable in the corresponding Word Game.



Conclusion and open questions

- Matrix Game is open for dimensions $2, 3$.
- Braid Game starting from particular word is completed.
  - B_2 is isomorphic to $(Z, +)$.
- Braid Game starting from trivial braid is open for B_3, B_4.
  - Same technique cannot be applied, as B_4 does not have direct product of two free groups.
- Application of the automaton to other games, models, etc ...

THANK YOU!