Improved Undecidability Results on the Emptiness Problem of Probabilistic and Quantum Cut-Point Languages

Mika Hirvensalo

mikhirve@utu.fi

Department of Mathematics

University of Turku

FIN-20014 Turku, Finland

Preliminaries

- *Stochastic* (*Markov*) matrix: Each column is a probability distribution.
- *Adjoint* matrix: $(M^*)_{ij} = (M_{ji})^*$.
- *Unitary* matrix: $U^*U = UU^* = I$.
- *Alphabet* $\Sigma = \{a_1, \ldots, a_k\}$ is a finite set.
- *Words* over Σ: Σ^*.

Preliminaries

- An n-state *probabilistic automaton* over Σ: $P = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$. $\boldsymbol{y} \in \mathbb{R}^n$ is the *initial distribution*, $\boldsymbol{x} \in \{0,1\}^n$ is the *final state vector*, and each M_a is an $n \times n$ stochastic matrix.

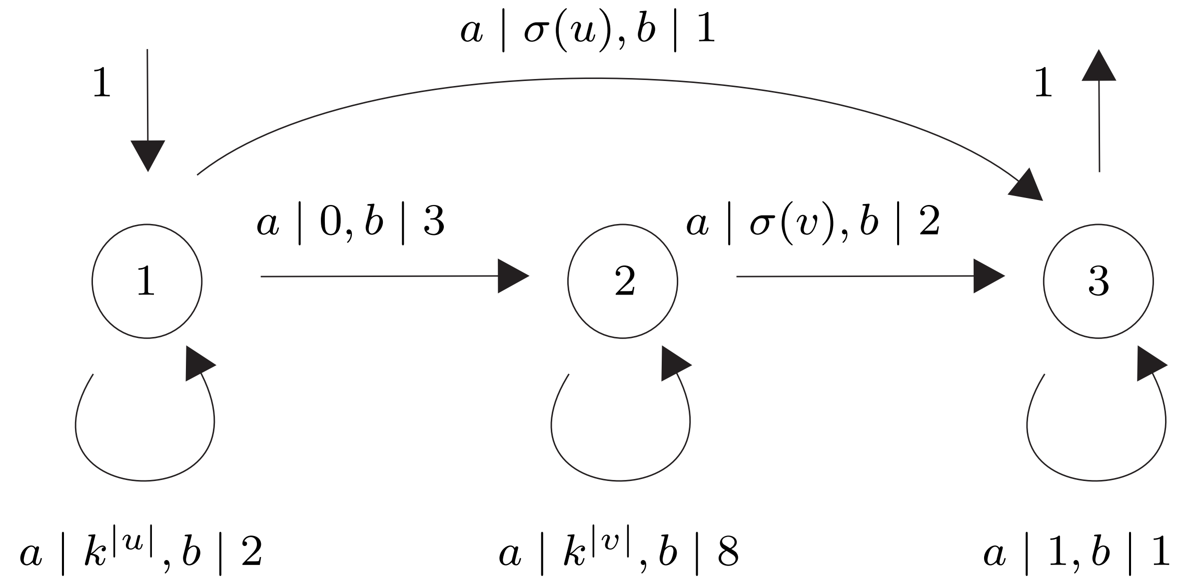
Preliminaries

- An n-state *probabilistic automaton* over Σ: $P = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$. $\boldsymbol{y} \in \mathbb{R}^n$ is the *initial distribution*, $\boldsymbol{x} \in \{0,1\}^n$ is the *final state vector*, and each M_a is an $n \times n$ stochastic matrix.
- An n-state *quantum automaton* over Σ: $Q = (P, \{U_a \mid a \in \Sigma\}, \boldsymbol{y})$. $\boldsymbol{y} \in \mathbb{C}^n$ is the *initial amplitude vector* with $||\boldsymbol{y}|| = 1$, P is the *measurement projection*, and each U_a is an $n \times n$ unitary matrix.

Preliminaries

- An n-state *quantum automaton* over Σ: $Q = (P, \{U_a \mid a \in \Sigma\}, \boldsymbol{y})$. $\boldsymbol{y} \in \mathbb{C}^n$ is the *initial amplitude vector* with $||\boldsymbol{y}|| = 1$, P is the *measurement projection*, and each U_a is an $n \times n$ unitary matrix.
- A $\mathbb{Z}$*-automaton*: matrices and vectors with integer entries.

Graph Representation



$$M_a = \begin{pmatrix} k^{|u|} & 0 & 0 \\ 0 & k^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix}, \quad M_b = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 8 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Preliminaries

- Let $P = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$ be a probabilistic automaton over Σ.

Preliminaries

- Let $P = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$ be a probabilistic automaton over Σ.
- For $w = a_1 \ldots a_r \in \Sigma^*$, $f_P(w)$ is defined as $f_P(w) = \boldsymbol{x}^T M_{a_r} \cdot \ldots \cdot M_{a_1} \boldsymbol{y}$. Analogously for $\mathbb{Z}$-automata.

Preliminaries

- Let $P = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$ be a probabilistic automaton over Σ.
- For $w = a_1 \ldots a_r \in \Sigma^*$, $f_P(w)$ is defined as $f_P(w) = \boldsymbol{x}^T M_{a_r} \cdot \ldots \cdot M_{a_1} \boldsymbol{y}$. Analogously for $\mathbb{Z}$-automata.
- Let $Q = (P, \{U_a \mid a \in \Sigma\}, \boldsymbol{y})$ be a quantum automaton over Σ.

Preliminaries

- Let $P = (\boldsymbol{x}, \{M_a \mid a \in \Sigma\}, \boldsymbol{y})$ be a probabilistic automaton over Σ.
- For $w = a_1 \ldots a_r \in \Sigma^*$, $f_P(w)$ is defined as $f_P(w) = \boldsymbol{x}^T M_{a_r} \cdot \ldots \cdot M_{a_1} \boldsymbol{y}$. Analogously for $\mathbb{Z}$-automata.
- Let $Q = (P, \{U_a \mid a \in \Sigma\}, \boldsymbol{y})$ be a quantum automaton over Σ.
- For $w = a_1 \ldots a_r \in \Sigma^*$, $f_Q(w)$ is defined as $f_Q(w) = ||PU_{a_r} \cdot \ldots \cdot U_{a_1} \boldsymbol{y}||^2$.

Cut-point languages

For any $\lambda \in [0,1]$ and automaton A let

$$L_{\geq\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) \geq \lambda\},$$

Cut-point languages

For any $\lambda \in [0, 1]$ and automaton A let

$$L_{\geq\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) \geq \lambda\},$$

a *cut-point language*

Cut-point languages

For any $\lambda \in [0,1]$ and automaton A let

$$L_{\geq\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) \geq \lambda\},$$

a *cut-point language*, and

$$L_{>\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) > \lambda\}$$

a *strict cut-point language*.

Cut-point languages

For any $\lambda \in [0,1]$ and automaton A let

$$L_{\geq\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) \geq \lambda\},$$

a *cut-point language*, and

$$L_{>\lambda}(A) = \{w \in \Sigma^* \mid f_A(w) > \lambda\}$$

a *strict cut-point language*.

The problems studied: given a binary automaton A and λ, is $L_{\geq\lambda}(A) = \emptyset$? Is $L_{>\lambda}(A) = \emptyset$?

Results

- $L_{\geq\lambda}(A) = \emptyset$?

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003).

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003). New : 25 states.

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003). New : 25 states.
- $L_{\geq\lambda}(A) = \emptyset$?

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003). New : 25 states.
- $L_{\geq\lambda}(A) = \emptyset$? undecidable for quantum automata with 43 states. (V. Blondel, E. Jeandel, P. Koiran, and N. Portier 2005).

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003). New : 25 states.
- $L_{\geq\lambda}(A) = \emptyset$? undecidable for quantum automata with 43 states. (V. Blondel, E. Jeandel, P. Koiran, and N. Portier 2005). New: 21 states.

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003). New : 25 states.
- $L_{\geq\lambda}(A) = \emptyset$? undecidable for quantum automata with 43 states. (V. Blondel, E. Jeandel, P. Koiran, and N. Portier 2005). New: 21 states.
- $L_{>\lambda}(A) = \emptyset$? undecidable for probabilistic automata.

Results

- $L_{\geq\lambda}(A) = \emptyset$? undecidable for probabilistic automata with 47 states. (V. Blondel and V. Canterini: 2003). New : 25 states.
- $L_{\geq\lambda}(A) = \emptyset$? undecidable for quantum automata with 43 states. (V. Blondel, E. Jeandel, P. Koiran, and N. Portier 2005). New: 21 states.
- $L_{>\lambda}(A) = \emptyset$? undecidable for probabilistic automata.
- $L_{>\lambda}(A) = \emptyset$? decidable for quantum automata.

Post Correspondence Problem (PCP)

Post Correspondence Problem (PCP)

Given pairs of words $\mathcal{I} = \{(u_1, v_1,), \ldots, (u_k, v_k)\}$, decide if there is a sequence of indices $i_1, \ldots, i_n$ so that $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$.

Post Correspondence Problem (PCP)

Given pairs of words $\mathcal{I} = \{(u_1, v_1,), \ldots, (u_k, v_k)\}$, decide if there is a sequence of indices $i_1, \ldots, i_n$ so that $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$.

- Undecidable for $k \geq 7$ (Y. Matiyasevich and G. Sénizergues 2005).

Post Correspondence Problem (PCP)

Given pairs of words $\mathcal{I} = \{(u_1, v_1,), \ldots, (u_k, v_k)\}$, decide if there is a sequence of indices $i_1, \ldots, i_n$ so that $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$.

- Undecidable for $k \geq 7$ (Y. Matiyasevich and G. Sénizergues 2005).
- Undecidable if all minimal solutions are of form $i_1 = 1$, $i_n = k$, and $i_2 \ldots i_{n-1} \in \{2, \ldots, k-1\}^+$. (V. Claus 1980).

Encodings

$$\sigma(i_1 i_2 \ldots i_n) = \sum_{j=1}^{n} i_j 2^{n-j}$$

Encodings

$$\sigma(i_1 i_2 \ldots i_n) = \sum_{j=1}^{n} i_j 2^{n-j}$$

- σ is a bijection $\Sigma^* = \{1, 2\}^* \to \mathbb{N}$.

Encodings

$$\sigma(i_1 i_2 \ldots i_n) = \sum_{j=1}^{n} i_j 2^{n-j}$$

- σ is a bijection $\Sigma^* = \{1,2\}^* \to \mathbb{N}$.
- $\sigma(1) = 1$, $\sigma(2) = 2$, $\sigma(11) = 3$, $\sigma(12) = 4$, $\sigma(21) = 5$, $\sigma(22) = 6$, $\sigma(111) = 7$, $\ldots$.

Encodings

$$\sigma(i_1 i_2 \ldots i_n) = \sum_{j=1}^{n} i_j 2^{n-j}$$

- σ is a bijection $\Sigma^* = \{1,2\}^* \to \mathbb{N}$.
- $\sigma(1) = 1$, $\sigma(2) = 2$, $\sigma(11) = 3$, $\sigma(12) = 4$, $\sigma(21) = 5$, $\sigma(22) = 6$, $\sigma(111) = 7$, $\ldots$.
- $\sigma(uv) = \sigma(u)2^{|v|} + \sigma(v)$.

Encodings

$$\sigma(i_1 i_2 \ldots i_n) = \sum_{j=1}^{n} i_j 2^{n-j}$$

- σ is a bijection $\Sigma^* = \{1,2\}^* \to \mathbb{N}$.
- $\sigma(1) = 1$, $\sigma(2) = 2$, $\sigma(11) = 3$, $\sigma(12) = 4$, $\sigma(21) = 5$, $\sigma(22) = 6$, $\sigma(111) = 7$, $\ldots$.
- $\sigma(uv) = \sigma(u)2^{|v|} + \sigma(v)$.

$$\delta(u) = \begin{pmatrix} 2^{|u|} & 0 \\ \sigma(u) & 1 \end{pmatrix}$$

Encodings

$$\delta(u) = \begin{pmatrix} 2^{|u|} & 0 \\ \sigma(u) & 1 \end{pmatrix}$$

Encodings

$$\delta(u) = \begin{pmatrix} 2^{|u|} & 0 \\ \sigma(u) & 1 \end{pmatrix}$$

- δ is an embedding $\Sigma^* \longrightarrow \mathbb{N}^{2\times 2}$ ($\delta(uv) = \delta(u)\delta(v)$).

Encodings

$$\delta(u) = \begin{pmatrix} 2^{|u|} & 0 \\ \sigma(u) & 1 \end{pmatrix}$$

- δ is an embedding $\Sigma^* \longrightarrow \mathbb{N}^{2\times 2}$ ($\delta(uv) = \delta(u)\delta(v)$).

$$\gamma_0(u, v) = \begin{pmatrix} 2^{|u|} & 0 & 0 \\ 0 & 2^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix}$$

- γ_0 is an embedding $\Sigma^* \times \Sigma^* \to \mathbb{N}^{3\times 3}$.

Encodings

$$\delta(u) = \begin{pmatrix} 2^{|u|} & 0 \\ \sigma(u) & 1 \end{pmatrix}$$

- δ is an embedding $\Sigma^* \longrightarrow \mathbb{N}^{2\times 2}$.

$$\gamma_0(u,v) = \begin{pmatrix} 2^{|u|} & 0 & 0 \\ 0 & 2^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix}$$

- γ_0 is an embedding $\Sigma^* \times \Sigma^* \longrightarrow \mathbb{N}^{3\times 3}$; $\gamma(u_1, v_1)\gamma(u_2, v_2) = \gamma(u_1 u_2, v_1 v_2)$.

Encodings

$$\gamma(u,v) = \begin{pmatrix} 2^{2|u|} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{|uv|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{2|v|} & 0 & 0 & 0 \\ \sigma(u)2^{|u|} & \sigma(v)2^{|u|} & 0 & 2^{|u|} & 0 & 0 \\ 0 & \sigma(u)2^{|v|} & \sigma(v)2^{|v|} & 0 & 2^{|v|} & 0 \\ \sigma(u)^2 & 2\sigma(u)\sigma(v) & \sigma(v)^2 & 2\sigma(u) & 2\sigma(v) & 1 \end{pmatrix}$$

- γ is an embedding $\Sigma^* \times \Sigma^* \to \mathbb{N}^{6\times 6}$.

Encodings

$$\gamma(u,v) = \begin{pmatrix} 2^{2|u|} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{|uv|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{2|v|} & 0 & 0 & 0 \\ \sigma(u)2^{|u|} & \sigma(v)2^{|u|} & 0 & 2^{|u|} & 0 & 0 \\ 0 & \sigma(u)2^{|v|} & \sigma(v)2^{|v|} & 0 & 2^{|v|} & 0 \\ \sigma(u)^2 & 2\sigma(u)\sigma(v) & \sigma(v)^2 & 2\sigma(u) & 2\sigma(v) & 1 \end{pmatrix}$$

- γ is an embedding $\Sigma^* \times \Sigma^* \to \mathbb{N}^{6\times 6}$.
- $\boldsymbol{x}_1^T \gamma(u,v) \boldsymbol{y}_1 = 1 - (\sigma(u) - \sigma(v))^2$ for $\boldsymbol{x}_1 = (0,0,0,0,0,1)^T$ and $\boldsymbol{y}_1 = (-1,1,-1,0,0,1)^T$.

Encodings

$$\gamma(u,v) = \begin{pmatrix} 2^{2|u|} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{|uv|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{2|v|} & 0 & 0 & 0 \\ \sigma(u)2^{|u|} & \sigma(v)2^{|u|} & 0 & 2^{|u|} & 0 & 0 \\ 0 & \sigma(u)2^{|v|} & \sigma(v)2^{|v|} & 0 & 2^{|v|} & 0 \\ \sigma(u)^2 & 2\sigma(u)\sigma(v) & \sigma(v)^2 & 2\sigma(u) & 2\sigma(v) & 1 \end{pmatrix}$$

- $\boldsymbol{x}_1^T \gamma(u,v) \boldsymbol{y}_1 = 1 - (\sigma(u) - \sigma(v))^2$ for $\boldsymbol{x}_1 = (0,0,0,0,0,1)^T$ and $\boldsymbol{y}_1 = (-1,1,-1,0,0,1)^T$.
- $\boldsymbol{x}_1^T \gamma(u,v) \boldsymbol{y}_1 > 0$ if and only if $u = v$.

From PCP to matrices

From PCP to matrices

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.

From PCP to matrices

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.

From PCP to matrices

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 = 1 - (\sigma(u_{i_1} \ldots u_{i_n}) - \sigma(v_{i_1} \ldots v_{i_n}))^2$.

From PCP to matrices

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 = 1 - (\sigma(u_{i_1} \ldots u_{i_n}) - \sigma(v_{i_1} \ldots v_{i_n}))^2$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 > 0$ if and only if $\mathcal{I}$ has a solution.

From PCP to matrices

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 = 1 - (\sigma(u_{i_1} \ldots u_{i_n}) - \sigma(v_{i_1} \ldots v_{i_n}))^2$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 > 0$ if and only if $\mathcal{I}$ has a solution.
- $f_A(w) > 0$ for some $w \in \Sigma^+$ undecidable for $\mathbb{Z}$-automata with 6 states and $k = 7$ alphabet symbols.

From PCP to matrices

- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 = 1 - (\sigma(u_{i_1} \ldots u_{i_n}) - \sigma(v_{i_1} \ldots v_{i_n}))^2$.
- $\boldsymbol{x}_1^T A_{i_1} \ldots A_{i_n} \boldsymbol{y}_1 > 0$ if and only if $\mathcal{I}$ has a solution.
- $f_A(w) > 0$ for some $w \in \Sigma^+$ undecidable for $\mathbb{Z}$-automata with 6 states and $k = 7$ alphabet symbols.
- A shorthand notation: For $w = i_1 \ldots i_n$, let $A_w = A_{i_1} \ldots A_{i_n}$.

Reducing the number of matrices

Reducing the number of matrices

- $\boldsymbol{x}_2 = (\boldsymbol{x}_1^T A_1)^T$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, $B_1 = A_2$, $\ldots$, $B_{k-2} = A_{k-1}$.

Reducing the number of matrices

- $\boldsymbol{x}_2 = (\boldsymbol{x}_1^T A_1)^T$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, $B_1 = A_2, \ldots,$ $B_{k-2} = A_{k-1}$.
- $\boldsymbol{x}_2^T B_w \boldsymbol{y}_2 = \boldsymbol{x}_1^T A_1 B_w A_k \boldsymbol{y}_1 > 0$ iff $\mathcal{I}$ has a solution (V. Claus).

Reducing the number of matrices

- $\boldsymbol{x}_2 = (\boldsymbol{x}_1^T A_1)^T$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, $B_1 = A_2, \ldots,$ $B_{k-2} = A_{k-1}$.
- $\boldsymbol{x}_2^T B_w \boldsymbol{y}_2 = \boldsymbol{x}_1^T A_1 B_w A_k \boldsymbol{y}_1 > 0$ iff $\mathcal{I}$ has a solution (V. Claus).
- $f_A(w) > 0$ for some $w \in \Sigma^+$ undecidable for $\mathbb{Z}$-automata with 6 states and $k - 2 = 5$ alphabet symbols.

Reducing the number of matrices

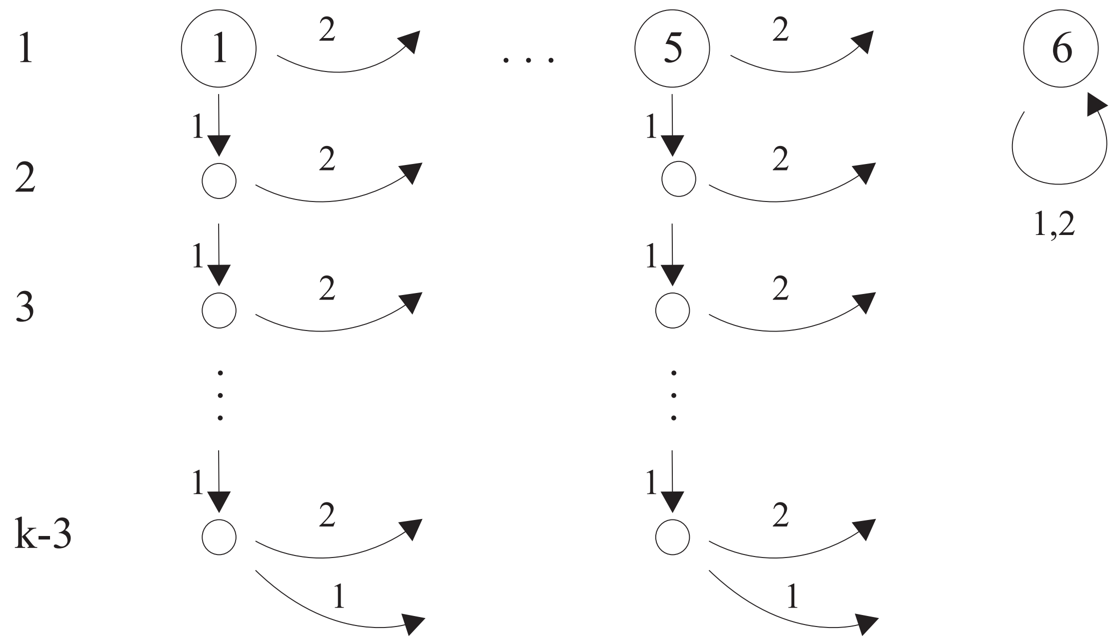
- $\boldsymbol{x}_2 = (\boldsymbol{x}_1^T A_1)^T$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, $B_1 = A_2, \ldots,$ $B_{k-2} = A_{k-1}$.
- $\boldsymbol{x}_2^T B_w \boldsymbol{y}_2 = \boldsymbol{x}_1^T A_1 B_w A_k \boldsymbol{y}_1 > 0$ iff $\mathcal{I}$ has a solution (V. Claus).
- $f_A(w) > 0$ for some $w \in \Sigma^+$ undecidable for $\mathbb{Z}$-automata with 6 states and $k-2=5$ alphabet symbols.
- For $k-2=5$ alphabet symbols we define $\psi(1) = 2$, $\psi(2) = 12$, $\psi(3) = 112$, $\psi(4) = 1112$, $\psi(5) = 1111$.

An observation

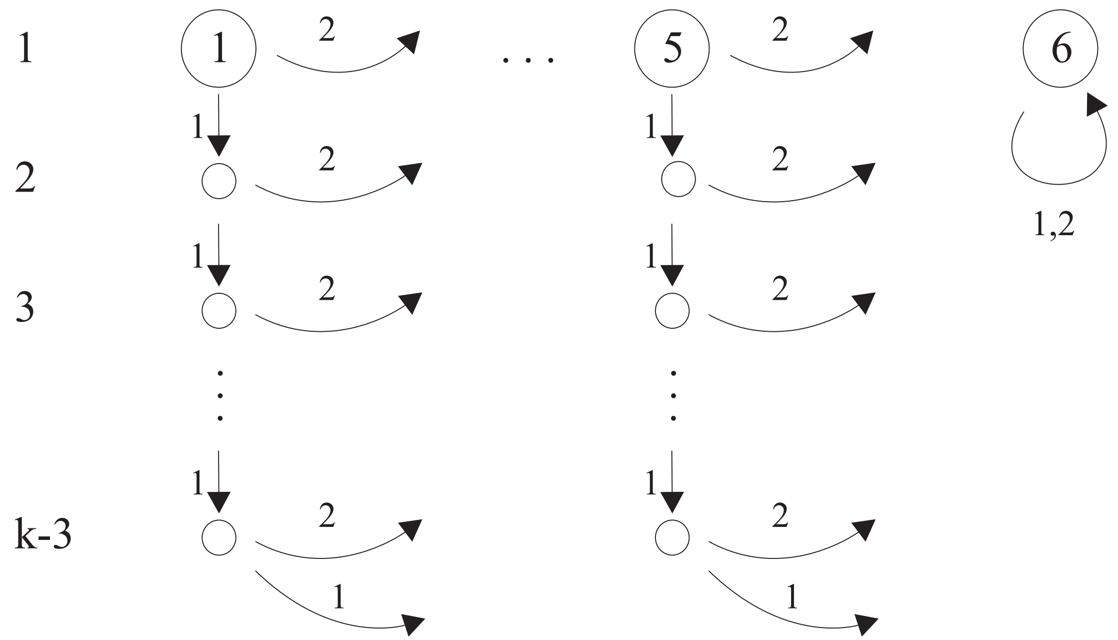
$$B_{i-1} = \gamma(u_i, v_i)$$

$$= \begin{pmatrix} 2^{2|u_i|} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{|u_i v_i|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{2|v_i|} & 0 & 0 & 0 \\ \sigma(u_i)2^{|u_i|} & \sigma(v_i)2^{|u_i|} & 0 & 2^{|u_i|} & 0 & 0 \\ 0 & \sigma(u_i)2^{|v_i|} & \sigma(v_i)2^{|v_i|} & 0 & 2^{|v_i|} & 0 \\ \sigma(u_i)^2 & 2\sigma(u_i)\sigma(v_i) & \sigma(v_i)^2 & 2\sigma(u_i) & 2\sigma(v_i) & \color{red}{1} \end{pmatrix}$$

Binary $\mathbb{Z}$-automaton

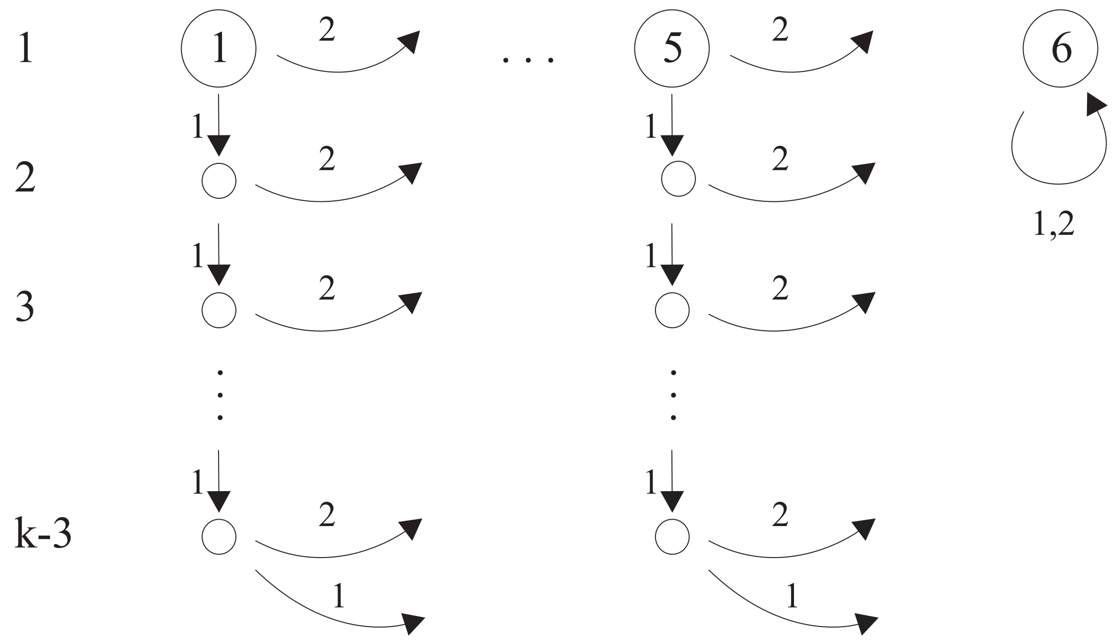


Binary $\mathbb{Z}$-automaton



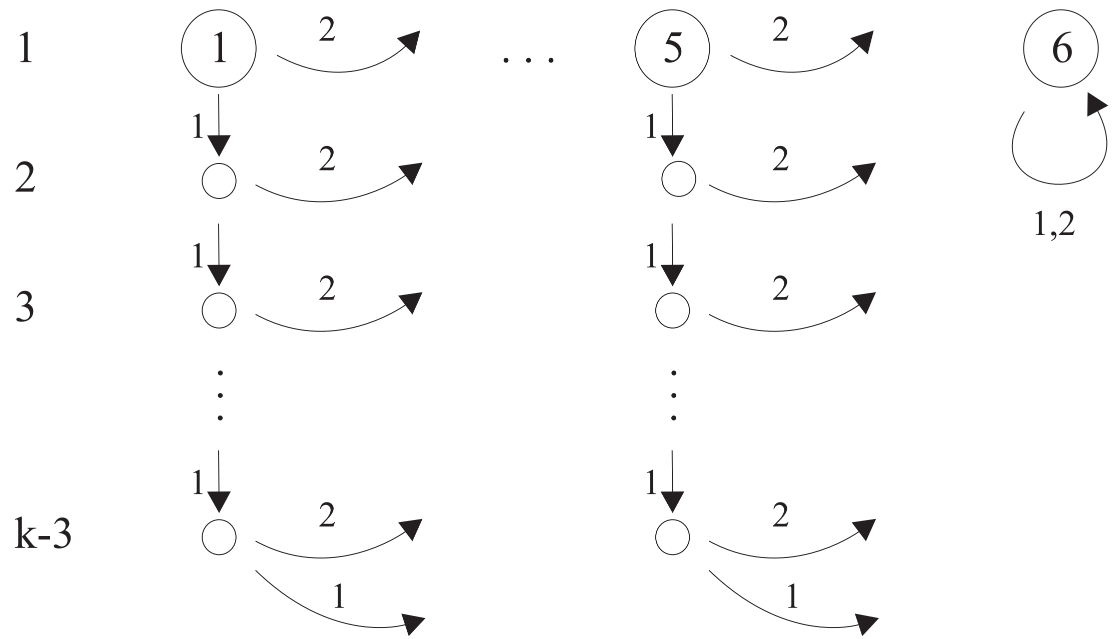
- The numbers are input symbols, not the weights.

Binary $\mathbb{Z}$-automaton



- The numbers are input symbols, not the weights.
- New automaton has $5(k-3)+1 = 5k-14 = 21$ states.

Binary $\mathbb{Z}$-automaton



- New automaton has $5(k-3)+1 = 5k-14 = 21$ states.
- $\mathcal{I}$ has a solution iff $\boldsymbol{x}_3^T C_w \boldsymbol{y}_3 > 0$ for some $w \in \{1,2\}^*$.

From $\mathbb{Z}$-automaton to probabilistic

- Procedure by P. Turakainen (1969)

From $\mathbb{Z}$-automaton to probabilistic

- Procedure by P. Turakainen (1969)

1. $$D_i = \begin{pmatrix} 0 & 0 & 0 \\ C_i \boldsymbol{y}_3 & C_i & 0 \\ \boldsymbol{x}_3^T C_i \boldsymbol{y}_3 & \boldsymbol{x}_3^T C_i & 0 \end{pmatrix}, \boldsymbol{x}_4 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

$$\boldsymbol{y}_4 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}; \boldsymbol{x}_4^T D_w \boldsymbol{y}_4 = \boldsymbol{x}_3^T C_w \boldsymbol{y}_3; 21 + 2 = 23$$

states.

From $\mathbb{Z}$-automaton to probabilistic

- Procedure by P. Turakainen (1969)

2. $E_i = \begin{pmatrix} 0 & 0 & 0 \\ \boldsymbol{t}_i & D_i & 0 \\ s_i & \boldsymbol{r}_i^T & 0 \end{pmatrix}$,

$\boldsymbol{x}_5 = (0, \boldsymbol{x}_4^T, 0)^T$, $\boldsymbol{y}_5 = (0, \boldsymbol{y}_4^T, 0)^T$;
$\boldsymbol{x}_5^T E_w \boldsymbol{y}_5 = \boldsymbol{x}_4^T D_w \boldsymbol{y}_4$; $23 + 2 = 25$ states.

From $\mathbb{Z}$-automaton to probabilistic

3. $F_i = E_i + c\mathbf{1}$. Note that $E_i\mathbf{1} = \mathbf{1}E_i = \mathbf{0}$, $\mathbf{1}^k = 25^{k-1}\mathbf{1}$; $F_w = E_w + c^{|w|}25^{|w|-1}\mathbf{1}$.

From $\mathbb{Z}$-automaton to probabilistic

3. $F_i = E_i + c\mathbf{1}$. Note that $E_i\mathbf{1} = \mathbf{1}E_i = \mathbf{0}$, $\mathbf{1}^k = 25^{k-1}\mathbf{1}$; $F_w = E_w + c^{|w|}25^{|w|-1}\mathbf{1}$.

4. $G_i = \frac{1}{25c}F_i$. Now $G_w = \frac{1}{(25c)^{|w|}}E_w + \frac{1}{25}\mathbf{1}$, and $\boldsymbol{x}_5^T G_w \boldsymbol{y}_5 = \frac{1}{(25c)^{|w|}}\boldsymbol{x}_5^T E_w \boldsymbol{y}_5 + \frac{1}{25}$. Notice also that G_1 and G_2 are (doubly) stochastic matrices.

From $\mathbb{Z}$-automaton to probabilistic

3. $F_i = E_i + c\mathbf{1}$. Note that $E_i\mathbf{1} = \mathbf{1}E_i = \mathbf{0}$, $\mathbf{1}^k = 25^{k-1}\mathbf{1}$; $F_w = E_w + c^{|w|}25^{|w|-1}\mathbf{1}$.

4. $G_i = \frac{1}{25c}F_i$. Now $G_w = \frac{1}{(25c)^{|w|}}E_w + \frac{1}{25}\mathbf{1}$, and $\boldsymbol{x}_5^T G_w \boldsymbol{y}_5 = \frac{1}{(25c)^{|w|}}\boldsymbol{x}_5^T E_w \boldsymbol{y}_5 + \frac{1}{25}$. Notice also that G_1 and G_2 are (doubly) stochastic matrices.

Theorem: For a 25-state probabilistic automaton $(\boldsymbol{x}_5, \{G_1, G_2\}, \boldsymbol{y}_5)$, $\boldsymbol{x}_5^T G_w \boldsymbol{y}_5 > \frac{1}{25}$ for some $w \in \Sigma^*$ if and only if $\mathcal{I}$ has a solution.

Quantum automata

Quantum automata

Let $U_1 = \frac{1}{5}\begin{pmatrix} 3 & -4 \\ 4 & 3 \end{pmatrix}$, $U_2 = \frac{1}{5}\begin{pmatrix} 3 & 4i \\ 4i & 3 \end{pmatrix}$ (both unitary), and $\boldsymbol{y} = (1, 0)^T$

Quantum automata

Let $U_1 = \frac{1}{5}\begin{pmatrix} 3 & -4 \\ 4 & 3 \end{pmatrix}$, $U_2 = \frac{1}{5}\begin{pmatrix} 3 & 4i \\ 4i & 3 \end{pmatrix}$ (both unitary), and $\boldsymbol{y} = (1, 0)^T$

Lemma: For each u, $v \in \Sigma^* = \{1, 2\}^*$, equality $U_u\boldsymbol{y} = U_v\boldsymbol{y}$ implies $u = v$.

Quantum automata

Define $\gamma(u,v) = \frac{1}{2}\begin{pmatrix} U_u + U_v & U_u - U_v \\ U_u - U_v & U_u + U_v \end{pmatrix}$,

$\boldsymbol{y}_1 = (\boldsymbol{y}, \boldsymbol{0})^T$, and $P_1 = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$.

Quantum automata

Define $\gamma(u,v) = \frac{1}{2}\begin{pmatrix} U_u + U_v & U_u - U_v \\ U_u - U_v & U_u + U_v \end{pmatrix}$,

$\boldsymbol{y}_1 = (\boldsymbol{y}, \mathbf{0})^T$, and $P_1 = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$.

$$P_1\gamma(u,v)\boldsymbol{y}_1 = \begin{pmatrix} \mathbf{0} \\ U_u\boldsymbol{y} - U_v\boldsymbol{y} \end{pmatrix}.$$

Quantum automata

Define $\gamma(u,v) = \frac{1}{2}\begin{pmatrix} U_u + U_v & U_u - U_v \\ U_u - U_v & U_u + U_v \end{pmatrix}$,

$\boldsymbol{y}_1 = (\boldsymbol{y}, \mathbf{0})^T$, and $P_1 = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$.

$$P_1\gamma(u,v)\boldsymbol{y}_1 = \begin{pmatrix} \mathbf{0} \\ U_u\boldsymbol{y} - U_v\boldsymbol{y} \end{pmatrix},$$

so $||P_1\gamma(u,v)\boldsymbol{y}_1||^2 = 0$ iff $u = v$.

From PCP to quantum automata

From PCP to quantum automata

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.

From PCP to quantum automata

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.

From PCP to quantum automata

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, …, $A_k = \gamma(u_k, v_k)$.
- $||P_1 A_w \boldsymbol{y}_1||^2 = 0$ for some nonempty w if and only if $\mathcal{I}$ has a solution.

From PCP to quantum automata

- Let $\mathcal{I} = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be an instance of PCP.
- Define $A_1 = \gamma(u_1, v_1)$, ..., $A_k = \gamma(u_k, v_k)$.
- $||P_1 A_w \boldsymbol{y}_1||^2 = 0$ for some nonempty w if and only if $\mathcal{I}$ has a solution.
- $\Rightarrow f_Q(w) = ||P_1 A_w \boldsymbol{y}_1||^2 = 0$ for some nonempty w is undecidable for quantum automata with 4 states and 7 alphabet symbols.

Reducing the number of matrices

Reducing the number of matrices

- Let $B_1 = A_2, \ldots, B_{k-2} = A_{k-1}$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, and $P_2 = A_1^{-1} P_1 A_1$.

Reducing the number of matrices

- Let $B_1 = A_2, \ldots, B_{k-2} = A_{k-1}$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, and $P_2 = A_1^{-1} P_1 A_1$.
- $||P_2 B_w \boldsymbol{y}_2||^2 = ||A_1^{-1} P_1 A_1 B_w A_k \boldsymbol{y}_1||^2 = ||P_1 A_1 B_w A_k \boldsymbol{y}_1||^2$

Reducing the number of matrices

- Let $B_1 = A_2, \ldots, B_{k-2} = A_{k-1}$, $\boldsymbol{y}_2 = A_k \boldsymbol{y}_1$, and $P_2 = A_1^{-1} P_1 A_1$.
- $||P_2 B_w \boldsymbol{y}_2||^2 = ||A_1^{-1} P_1 A_1 B_w A_k \boldsymbol{y}_1||^2 = ||P_1 A_1 B_w A_k \boldsymbol{y}_1||^2$
- $\Rightarrow f_Q(w) = 0$ for some w is undecidable for quantum automata with 4 states and 5 alphabet symbols.

Reducing the number of matrices

$$C_1 = \begin{pmatrix} B_1 & 0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 & 0 \\ 0 & 0 & B_3 & 0 & 0 \\ 0 & 0 & 0 & B_4 & 0 \\ 0 & 0 & 0 & 0 & B_5 \end{pmatrix}, C_2 = \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & 0 \end{pmatrix}.$$

C_1 and C_2 are unitary 20×20-matrices. Let also

$$P_3 = \begin{pmatrix} P_2 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_2 \end{pmatrix}, \boldsymbol{y}_3 = \begin{pmatrix} \boldsymbol{y}_2 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

Reducing the number of matrices

$$C_1 = \begin{pmatrix} B_1 & 0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 & 0 \\ 0 & 0 & B_3 & 0 & 0 \\ 0 & 0 & 0 & B_4 & 0 \\ 0 & 0 & 0 & 0 & B_5 \end{pmatrix}, C_2 = \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- For each $w \in \{1,2\}^*$ $||P_3 C_w \boldsymbol{y}_3||^2 = ||P_2 B_{w'} \boldsymbol{y}_2||^2$ for some $w' \in \{1,\dots,5\}^*$.

Reducing the number of matrices

$$C_1 = \begin{pmatrix} B_1 & 0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 & 0 \\ 0 & 0 & B_3 & 0 & 0 \\ 0 & 0 & 0 & B_4 & 0 \\ 0 & 0 & 0 & 0 & B_5 \end{pmatrix}, C_2 = \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- For each $w \in \{1,2\}^*$ $||P_3 C_w \boldsymbol{y}_3||^2 = ||P_2 B_{w'} \boldsymbol{y}_2||^2$ for some $w' \in \{1, \ldots, 5\}^*$.
- $C_2 C_1 C_2^{-1} = Diag(B_2, B_3, \ldots, B_5, B_1)$

Reducing the number of matrices

$$C_1 = \begin{pmatrix} B_1 & 0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 & 0 \\ 0 & 0 & B_3 & 0 & 0 \\ 0 & 0 & 0 & B_4 & 0 \\ 0 & 0 & 0 & 0 & B_5 \end{pmatrix}, C_2 = \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- For each $w \in \{1,2\}^* \ ||P_3 C_w \boldsymbol{y}_3||^2 = ||P_2 B_{w'} \boldsymbol{y}_2||^2$ for some $w' \in \{1,\ldots,5\}^*$.
- $C_2 C_1 C_2^{-1} = Diag(B_2, B_3, \ldots, B_5, B_1)$
- $\Rightarrow \forall w \in \{1,\ldots,5\}^* \ ||P_2 B_w \boldsymbol{y}_2||^2 = ||P_3 C_{w'} \boldsymbol{y}_3||^2$ for some $w' \in \{1,2\}^*$.

Reducing the number of matrices

- For each $w \in \{1,2\}^* \; ||P_3 C_w \boldsymbol{y}_3||^2 = ||P_2 B_{w'} \boldsymbol{y}_2||^2$ for some $w' \in \{1,\ldots,5\}^*$.
- $C_2 C_1 C_2^{-1} = Diag(B_2, B_3, \ldots, B_5, B_1)$
- $\Rightarrow \forall w \in \{1,\ldots,5\}^* \; ||P_2 B_w \boldsymbol{y}_2||^2 = ||P_3 C_{w'} \boldsymbol{y}_3||^2$ for some $w' \in \{1,2\}^*$.

Reducing the number of matrices

- For each $w \in \{1,2\}^*$ $||P_3 C_w \boldsymbol{y}_3||^2 = ||P_2 B_{w'} \boldsymbol{y}_2||^2$ for some $w' \in \{1,\ldots,5\}^*$.
- $C_2 C_1 C_2^{-1} = Diag(B_2, B_3, \ldots, B_5, B_1)$
- $\Rightarrow \forall w \in \{1,\ldots,5\}^*$ $||P_2 B_w \boldsymbol{y}_2||^2 = ||P_3 C_{w'} \boldsymbol{y}_3||^2$ for some $w' \in \{1,2\}^*$.
- $\Rightarrow$ $||P_3 C_w \boldsymbol{y}_3||^2 = 0$ for some $w \in \{1,2\}^*$ if and only if $\mathcal{I}$ has a solution.

Reducing the number of matrices

- For each $w \in \{1,2\}^*$ $||P_3 C_w \boldsymbol{y}_3||^2 = ||P_2 B_{w'} \boldsymbol{y}_2||^2$ for some $w' \in \{1,\ldots,5\}^*$.
- $C_2 C_1 C_2^{-1} = Diag(B_2, B_3, \ldots, B_5, B_1)$
- $\Rightarrow \forall w \in \{1,\ldots,5\}^*$ $||P_2 B_w \boldsymbol{y}_2||^2 = ||P_3 C_{w'} \boldsymbol{y}_3||^2$ for some $w' \in \{1,2\}^*$.
- $\Rightarrow ||P_3 C_w \boldsymbol{y}_3||^2 = 0$ for some $w \in \{1,2\}^*$ if and only if $\mathcal{I}$ has a solution.
- $\Rightarrow f_Q(w) = 0$ for some w is undecidable for binary quantum automata with 20 states.

Setting the threshold

Setting the threshold

- $1 = ||C_w \boldsymbol{y}_3||^2 = ||(I - P_3)C_w \boldsymbol{y}_3||^2 + ||P_3 C_w \boldsymbol{y}_3||^2$, hence $||(I - P_3)C_w \boldsymbol{y}_3||^2 \geq 1$ if and only if $\mathcal{I}$ has a solution.

Setting the threshold

- $1 = ||C_w \boldsymbol{y}_3||^2 = ||(I - P_3)C_w \boldsymbol{y}_3||^2 + ||P_3 C_w \boldsymbol{y}_3||^2$, hence $||(I - P_3)C_w \boldsymbol{y}_3||^2 \geq 1$ if and only if $\mathcal{I}$ has a solution.
- Let $D_i = \begin{pmatrix} C_i & 0 \\ 0 & 1 \end{pmatrix}$, $P_4 = \begin{pmatrix} I - P_3 & 0 \\ 0 & 0 \end{pmatrix}$, and $\boldsymbol{y}_4 = (\sqrt{\lambda}\boldsymbol{y}_3^T, \sqrt{1-\lambda})^T$.

Setting the threshold

- $1 = ||C_w \boldsymbol{y}_3||^2 = ||(I - P_3)C_w \boldsymbol{y}_3||^2 + ||P_3 C_w \boldsymbol{y}_3||^2$, hence $||(I - P_3)C_w \boldsymbol{y}_3||^2 \geq 1$ if and only if $\mathcal{I}$ has a solution.
- Let $D_i = \begin{pmatrix} C_i & 0 \\ 0 & 1 \end{pmatrix}$, $P_4 = \begin{pmatrix} I - P_3 & 0 \\ 0 & 0 \end{pmatrix}$, and $\boldsymbol{y}_4 = (\sqrt{\lambda}\boldsymbol{y}_3^T, \sqrt{1-\lambda})^T$.
- $||P_4 D_w \boldsymbol{y}_4||^2 = ||\sqrt{\lambda}(I - P_3)C_w \boldsymbol{y}_4||^2 = \lambda(1 - ||P_3 C_w \boldsymbol{y}_3||^2)$.

Setting the threshold

- $1 = ||C_w \boldsymbol{y}_3||^2 = ||(I - P_3)C_w \boldsymbol{y}_3||^2 + ||P_3 C_w \boldsymbol{y}_3||^2$, hence $||(I - P_3)C_w \boldsymbol{y}_3||^2 \geq 1$ if and only if $\mathcal{I}$ has a solution.
- Let $D_i = \begin{pmatrix} C_i & 0 \\ 0 & 1 \end{pmatrix}$, $P_4 = \begin{pmatrix} I - P_3 & 0 \\ 0 & 0 \end{pmatrix}$, and $\boldsymbol{y}_4 = (\sqrt{\lambda}\boldsymbol{y}_3^T, \sqrt{1-\lambda})^T$.
- $||P_4 D_w \boldsymbol{y}_4||^2 = ||\sqrt{\lambda}(I - P_3)C_w \boldsymbol{y}_4||^2 = \lambda(1 - ||P_3 C_w \boldsymbol{y}_3||^2)$.
- $\Rightarrow f_Q(w) \geq \lambda$ for some some w is undecidable for binary quantum automata with 21 states.

Thank You!