Undecidability in automaton groups

P. Gillibert

Technische Universität Wien

February 2019

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

- We extend $\sigma\colon A^* \times \Sigma^* \to \Sigma^*$ and $\delta\colon A^* \times \Sigma^* \to A^*$.

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

- We extend $\sigma\colon A^* \times \Sigma^* \to \Sigma^*$ and $\delta\colon A^* \times \Sigma^* \to A^*$.
- With :

$$\sigma_u(xy) = \sigma_u(x)\sigma_{\delta_x(u)}(y)\,, \quad \text{for all } u \in A^*,\ x, y \in \Sigma^*.$$

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

- We extend $\sigma \colon A^* \times \Sigma^* \to \Sigma^*$ and $\delta \colon A^* \times \Sigma^* \to A^*$.
- With :

$$\sigma_u(xy) = \sigma_u(x)\sigma_{\delta_x(u)}(y)\,, \quad \text{for all } u \in A^*,\ x, y \in \Sigma^*.$$

$$\delta_x(uv) = \delta_x(u)\delta_{\sigma_u(x)}(v)\,, \quad \text{for all } x \in \Sigma^*,\ u, v \in A^*.$$

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

- We extend $\sigma\colon A^* \times \Sigma^* \to \Sigma^*$ and $\delta\colon A^* \times \Sigma^* \to A^*$.
- With :

$$\sigma_u(xy) = \sigma_u(x)\sigma_{\delta_x(u)}(y)\,, \quad \text{for all } u \in A^*,\ x, y \in \Sigma^*.$$

$$\delta_x(uv) = \delta_x(u)\delta_{\sigma_u(x)}(v)\,, \quad \text{for all } x \in \Sigma^*,\ u, v \in A^*.$$

$$\sigma_{uv} = \sigma_v \circ \sigma_u\,, \quad \text{for all } u, v \in A^*.$$

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

- We extend $\sigma : A^* \times \Sigma^* \to \Sigma^*$ and $\delta : A^* \times \Sigma^* \to A^*$.
- With :

$$\sigma_u(xy) = \sigma_u(x)\sigma_{\delta_x(u)}(y)\,, \quad \text{for all } u \in A^*,\ x, y \in \Sigma^*.$$

$$\delta_x(uv) = \delta_x(u)\delta_{\sigma_u(x)}(v)\,, \quad \text{for all } x \in \Sigma^*,\ u, v \in A^*.$$

$$\sigma_{uv} = \sigma_v \circ \sigma_u\,, \quad \text{for all } u, v \in A^*.$$

$$\delta_{xy} = \delta_y \circ \delta_x\,, \quad \text{for all } x, y \in \Sigma^*.$$

Mealy automaton

Given $\mathcal{A} = (A, \Sigma, \delta, \sigma)$ a Mealy automaton.

- We extend $\sigma\colon A^* \times \Sigma^* \to \Sigma^*$ and $\delta\colon A^* \times \Sigma^* \to A^*$.
- With :

$$\sigma_u(xy) = \sigma_u(x)\sigma_{\delta_x(u)}(y)\,, \quad \text{for all } u \in A^*,\ x, y \in \Sigma^*.$$

$$\delta_x(uv) = \delta_x(u)\delta_{\sigma_u(x)}(v)\,, \quad \text{for all } x \in \Sigma^*,\ u, v \in A^*.$$

$$\sigma_{uv} = \sigma_v \circ \sigma_u\,, \quad \text{for all } u, v \in A^*.$$

$$\delta_{xy} = \delta_y \circ \delta_x\,, \quad \text{for all } x, y \in \Sigma^*.$$

- Set $G(\mathcal{A})$ the group generated by $\{\sigma_u \mid u \in A\}$.

Cross diagrams

We use cross-diagrams to simplify notations. For all $u \in A^*$ and all $x \in \Sigma^*$.

$$\begin{array}{ccc} & x & \\ u & \text{---}\!\!\!\!+\!\!\!\!\longrightarrow & \delta_x(u) \\ & \downarrow & \\ & \sigma_u(x) & \end{array}$$

We use cross-diagrams to simplify notations. For all $u \in A^*$ and all $x \in \Sigma^*$.

$$\begin{tikzcd} & x \arrow[d] & \\ u \arrow[rr] & {} & \delta_x(u) \\ & \sigma_u(x) & \end{tikzcd}$$

The previous equalities are summarized by the following cross diagrams

Cross diagrams

We use cross-diagrams to simplify notations. For all $u \in A^*$ and all $x \in \Sigma^*$.

$$\begin{tikzcd} & x \arrow[dd] & \\ u \arrow[rr] & & \delta_x(u) \\ & \sigma_u(x) & \end{tikzcd}$$

The previous equalities are summarized by the following cross diagrams

$$\begin{tikzcd} & x \arrow[dd] & \\ u \arrow[rr] & & \delta_x(u) \\ & \sigma_u(x) \arrow[dd] & \\ v \arrow[rr] & & \delta_{\sigma_u(x)}(v) \\ & \sigma_{uv}(x) & \end{tikzcd}
\qquad
\begin{tikzcd} & x \arrow[dd] & & y \arrow[dd] & \\ u \arrow[rr] & & \delta_x(u) \arrow[rr] & & \delta_{xy}(u) \\ & \sigma_u(x) & & \sigma_{\delta_x(u)}(y) & \end{tikzcd}$$

Grigorchuk's Group

Grigorchuk's construted an automaton group G, with nice properties.

Grigorchuk’s Group

Grigorchuk’s construted an automaton group G, with nice properties.

- G is a Burnside 2-group.

Grigorchuk's Group

Grigorchuk's construted an automaton group G, with nice properties.

- G is a Burnside 2-group.
- G is of intermediate growth (neither polynomial, nor exponential), solving a problem by Milnor.

Grigorchuk's Group

Grigorchuk's construted an automaton group G, with nice properties.

- G is a Burnside 2-group.
- G is of intermediate growth (neither polynomial, nor exponential), solving a problem by Milnor.
- G is amenable but not elementary amenable (solving a question by Day)

Grigorchuk's Group

Grigorchuk's construted an automaton group G, with nice properties.

- G is a Burnside 2-group.
- G is of intermediate growth (neither polynomial, nor exponential), solving a problem by Milnor.
- G is amenable but not elementary amenable (solving a question by Day)
- G is just infinite.

Undecidability

- The word problem is decidable in automaton groups (Eilenberg's reduction algorithm).

Undecidability

- The word problem is decidable in automaton groups (Eilenberg's reduction algorithm).
- There is an automaton group with undecidable conjugacy problem (Šunić and Ventura).

Undecidability

- The word problem is decidable in automaton groups (Eilenberg's reduction algorithm).
- There is an automaton group with undecidable conjugacy problem (Šunić and Ventura).
- The finiteness problem for reset automaton semigroups is undecidable (G.).

Undecidability

- The word problem is decidable in automaton groups (Eilenberg's reduction algorithm).
- There is an automaton group with undecidable conjugacy problem (Šunić and Ventura).
- The finiteness problem for reset automaton semigroups is undecidable (G.).
- The freeness problem for automaton semigroups is undecidable (D'Angeli, Rodaro, and Wächter).

- The word problem is decidable in automaton groups (Eilenberg's reduction algorithm).
- There is an automaton group with undecidable conjugacy problem (Šunić and Ventura).
- The finiteness problem for reset automaton semigroups is undecidable (G.).
- The freeness problem for automaton semigroups is undecidable (D'Angeli, Rodaro, and Wächter).
- There is an automaton group with undecidable order problem (G.).

- Fix a Turing Machine $\mathcal{T}$. Encode it in a cellular automaton with alphabet T.

Computing with automaton groups

- Fix a Turing Machine $\mathcal{T}$. Encode it in a cellular automaton with alphabet T.
- The naïve approach...

Computing with automaton groups

- Fix a Turing Machine $\mathcal{T}$. Encode it in a cellular automaton with alphabet T.
- The naïve approach... Fails.
- Set $\Sigma = (T \sqcup \{\$\}) \times \{0, 1\}$.

Computing with automaton groups

- Fix a Turing Machine $\mathcal{T}$. Encode it in a cellular automaton with alphabet T.
- The naïve approach... Fails.
- Set $\Sigma = (T \sqcup \{\$\}) \times \{0, 1\}$.
- Set $A = \{\neg\} \sqcup \{$a bunch of counters and other states$\}$.

Computing with automaton groups

- Fix a Turing Machine $\mathcal{T}$. Encode it in a cellular automaton with alphabet T.
- The naïve approach... Fails.
- Set $\Sigma = (T \sqcup \{\$\}) \times \{0, 1\}$.
- Set $A = \{\neg\} \sqcup \{$a bunch of counters and other states$\}$.
- All states will either acts like identity or switch 0 and 1.
- $\neg$ will always switch 0 and 1.

Element of finite order

Set $Q(u_1 \dots u_n) = Q(u_1 \dots u_{n-1}) u_n Q(u_1 \dots u_{n-1})$.

Element of finite order

Set $Q(u_1 \dots u_n) = Q(u_1 \dots u_{n-1}) u_n Q(u_1 \dots u_{n-1})$.

Theorem

Let $a_1 \dots a_n \in T^*$. The following statement are equivalent

1. *After N steps the configuration $a_1 \dots a_n$ reach a final state.*

Element of finite order

Set $Q(u_1 \dots u_n) = Q(u_1 \dots u_{n-1}) u_n Q(u_1 \dots u_{n-1})$.

Theorem

Let $a_1 \dots a_n \in T^*$. The following statement are equivalent

1. *After N steps the configuration $a_1 \dots a_n$ reach a final state.*
2. *The element $\sigma_{Q(C^{a_1} \dots C^{a_n}) C^{\$}}$ is of finite order 2^{N+1}.*

Element of finite order

Set $Q(u_1 \dots u_n) = Q(u_1 \dots u_{n-1}) u_n Q(u_1 \dots u_{n-1})$.

Theorem

Let $a_1 \dots a_n \in T^*$. The following statement are equivalent

1. *After N steps the configuration $a_1 \dots a_n$ reach a final state.*
2. *The element $\sigma_{Q(C^{a_1} \dots C^{a_n}) C^{\$}}$ is of finite order 2^{N+1}.*

In particular if the Turing Machine is universal we deduce the following.

Element of finite order

Set $Q(u_1 \dots u_n) = Q(u_1 \dots u_{n-1}) u_n Q(u_1 \dots u_{n-1})$.

Theorem

Let $a_1 \dots a_n \in T^*$. The following statement are equivalent

1. *After N steps the configuration $a_1 \dots a_n$ reach a final state.*
2. *The element $\sigma_{Q(C^{a_1} \dots C^{a_n})C^{\$}}$ is of finite order 2^{N+1}.*

In particular if the Turing Machine is universal we deduce the following.

Theorem

There is an Automaton group G such that it is undecidable whether or not an element is of finite order.

That is all

Thanks.