Representations and Complexity of Abelian Automaton Groups

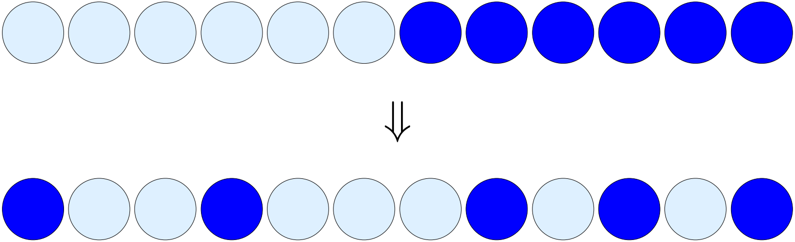
Tim Becker
Advised by: Klaus Sutner

Carnegie Mellon University

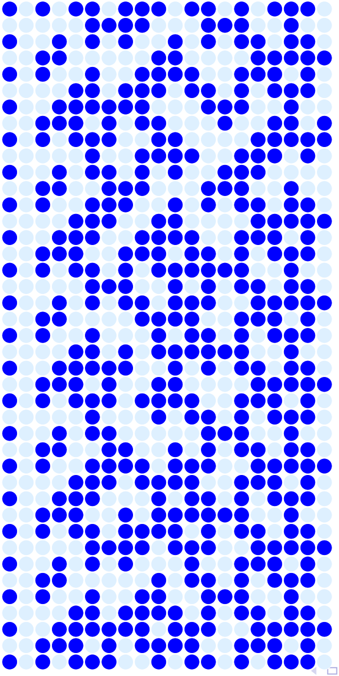
May 9, 2018

Table of Contents

Flipping Pebbles

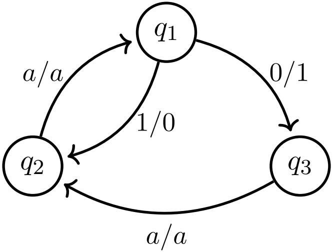


Rule: Given a row of pebbles, white on one side, blue on the other. Flip the first pebble. If it was white, skip the next two pebbles; otherwise, skip just one pebble. Keep flipping till you fall off the end.



Invertible Transducers

The previous game is described by a deterministic finite state transducer with invertible output functions.



Automaton Groups

- States induce a length preserving invertible function on binary strings.

Automaton Groups

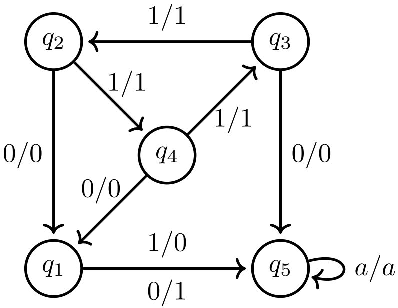
- States induce a length preserving invertible function on binary strings.
- These functions form a group under composition, called an automaton group.

Automaton Groups

- States induce a length preserving invertible function on binary strings.
- These functions form a group under composition, called an automaton group.
- These have been useful for some recent results in group theory.

Automaton Groups

- States induce a length preserving invertible function on binary strings.
- These functions form a group under composition, called an automaton group.
- These have been useful for some recent results in group theory.



Terminology and Notation

- $\mathbf{2} = \{0, 1\}$ is the binary alphabet.

Terminology and Notation

- $\mathbf{2} = \{0, 1\}$ is the binary alphabet.
- $\mathcal{A}$ is an invertible transducer.

Terminology and Notation

- $\mathbf{2} = \{0, 1\}$ is the binary alphabet.
- $\mathcal{A}$ is an invertible transducer.
- $\mathcal{A}^{-1}$ is the inverse machine, formed by flipping the edge labels:

$$q_i \xrightarrow{\mathbf{a/b}} q_j \qquad \text{becomes} \qquad q_i^{-1} \xrightarrow{\mathbf{b/a}} q_j^{-1}.$$

Terminology and Notation

- $\mathbf{2} = \{0, 1\}$ is the binary alphabet.
- $\mathcal{A}$ is an invertible transducer.
- $\mathcal{A}^{-1}$ is the inverse machine, formed by flipping the edge labels:

$$q_i \xrightarrow{\mathbf{a}/\mathbf{b}} q_j \qquad \text{becomes} \qquad q_i^{-1} \xrightarrow{\mathbf{b}/\mathbf{a}} q_j^{-1}.$$

- $\mathcal{G}(\mathcal{A})$ is the group formed by arbitrary compositions of transductions from $\mathcal{A}$ and $\mathcal{A}^{-1}$.

Terminology and Notation

- $\mathbf{2} = \{0, 1\}$ is the binary alphabet.
- $\mathcal{A}$ is an invertible transducer.
- $\mathcal{A}^{-1}$ is the inverse machine, formed by flipping the edge labels:

$$q_i \xrightarrow{\mathbf{a/b}} q_j \qquad \text{becomes} \qquad q_i^{-1} \xrightarrow{\mathbf{b/a}} q_j^{-1}.$$

- $\mathcal{G}(\mathcal{A})$ is the group formed by arbitrary compositions of transductions from $\mathcal{A}$ and $\mathcal{A}^{-1}$.
- An element of $\mathcal{G}(\mathcal{A})$ is even if it copies the first input bit, and odd if it flips it.

Residuals

Any $f \in \mathcal{G}(\mathcal{A})$ is a word over the states of $\mathcal{A}$:

$$f = q_{i_1}^{d_1}\ q_{i_2}^{d_2} \cdots q_{i_n}^{d_n},$$

where each $d_i \in \{-1, 1\}$.

Residuals

Any $f \in \mathcal{G}(\mathcal{A})$ is a word over the states of $\mathcal{A}$:

$$f = q_{i_1}^{d_1}\ q_{i_2}^{d_2} \cdots q_{i_n}^{d_n},$$

where each $d_i \in \{-1, 1\}$.
Hence f can be seen as a state in the product automaton

$$\mathcal{P} = \mathcal{A}^{d_1} \times \mathcal{A}^{d_2} \times \cdots \times \mathcal{A}^{d_n}.$$

Residuals

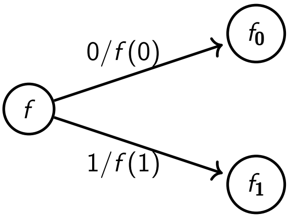
Any $f \in \mathcal{G}(\mathcal{A})$ is a word over the states of $\mathcal{A}$:

$$f = q_{i_1}^{d_1} \; q_{i_2}^{d_2} \cdots q_{i_n}^{d_n},$$

where each $d_i \in \{-1, 1\}$.
Hence f can be seen as a state in the product automaton

$$\mathcal{P} = \mathcal{A}^{d_1} \times \mathcal{A}^{d_2} \times \cdots \times \mathcal{A}^{d_n}.$$

For $\mathbf{a} \in \mathbf{2}$, then $f_{\mathbf{a}}$ is the **a**-residual of f: the state in $\mathcal{P}$ which f transitions to on input letter $\mathbf{a}$.

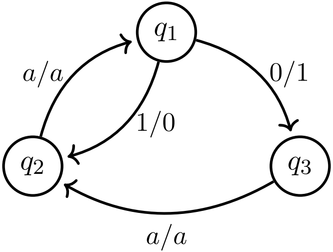


Abelian Automata

We will focus on machines $\mathcal{A}$ where $\mathcal{G}(\mathcal{A})$ is abelian, i.e. where all transductions commute.

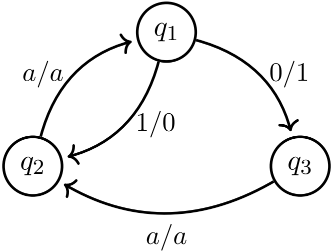
Abelian Automata

We will focus on machines $\mathcal{A}$ where $\mathcal{G}(\mathcal{A})$ is abelian, i.e. where all transductions commute.



Abelian Automata

We will focus on machines $\mathcal{A}$ where $\mathcal{G}(\mathcal{A})$ is abelian, i.e. where all transductions commute.



As functions over binary strings, we have $q_i q_j = q_j q_i$ for $i, j \in \{1, 2, 3\}$. Hence $\mathcal{G}(\mathcal{A})$ is abelian.

Gap Lemma

Define the gap value of any $f \in \mathcal{G}(\mathcal{A})$ as $\gamma_f = f_{\mathbf{0}} f_{\mathbf{1}}^{-1}$.

Gap Lemma

Define the gap value of any $f \in \mathcal{G}(\mathcal{A})$ as $\gamma_f = f_{\mathbf{0}} {f_{\mathbf{1}}}^{-1}$.

Gap Lemma: $\mathcal{G}(\mathcal{A})$ is abelian if and only if every odd element has the same gap value and every even element has gap value equal to the identity function.

Gap Lemma

Thus in an abelian automaton, there is a global gap value γ such that for all odd $f \in \mathcal{G}(\mathcal{A})$, $f_{\mathbf{0}} = \gamma f_{\mathbf{1}}$. Also if g is even, then $g_{\mathbf{0}} = g_{\mathbf{1}}$.

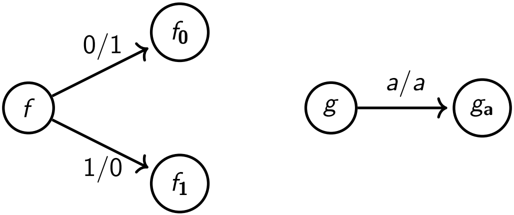


Table of Contents

Orbits

The orbit of a string $\mathbf{x} \in \mathbf{2}^k$ under f is

$$\{\mathbf{y} \in \mathbf{2}^k \mid \exists t \in \mathbb{Z} \text{ such that } f^t(\mathbf{x})\}.$$

Orbits

The orbit of a string $\mathbf{x} \in \mathbf{2}^k$ under f is

$$\{\mathbf{y} \in \mathbf{2}^k \mid \exists t \in \mathbb{Z} \text{ such that } f^t(\mathbf{x})\}.$$

The orbit language of f is

$$\mathbf{orb}(f) = \{\mathbf{x}{:}\mathbf{y} \mid \exists t \in \mathbb{Z} \text{ such that } f^t(\mathbf{x}) = \mathbf{y}\}$$

where $\mathbf{x}{:}\mathbf{y}$ is the convolution two words $\mathbf{x}, \mathbf{y} \in \mathbf{2}^k$:

$$\mathbf{x}{:}\mathbf{y} = \begin{array}{|c|c|c|c|} \hline \mathbf{x_1} & \mathbf{x_2} & \dots & \mathbf{x_k} \\ \hline \mathbf{y_1} & \mathbf{y_2} & \dots & \mathbf{y_k} \\ \hline \end{array} \in (\mathbf{2} \times \mathbf{2})^k.$$

Orbits

The orbit of a string $\mathbf{x} \in \mathbf{2}^k$ under f is

$$\{\mathbf{y} \in \mathbf{2}^k \mid \exists t \in \mathbb{Z} \text{ such that } f^t(\mathbf{x})\}.$$

The orbit language of f is

$$\mathbf{orb}(f) = \{\mathbf{x}{:}\mathbf{y} \mid \exists t \in \mathbb{Z} \text{ such that } f^t(\mathbf{x}) = \mathbf{y}\}$$

where $\mathbf{x}{:}\mathbf{y}$ is the convolution two words $\mathbf{x}, \mathbf{y} \in \mathbf{2}^k$:

$$\mathbf{x}{:}\mathbf{y} = \begin{array}{|c|c|c|c|} \hline \mathbf{x_1} & \mathbf{x_2} & \dots & \mathbf{x_k} \\ \hline \mathbf{y_1} & \mathbf{y_2} & \dots & \mathbf{y_k} \\ \hline \end{array} \in (\mathbf{2} \times \mathbf{2})^k.$$

Orbit Problem: Given $f \in \mathcal{G}(\mathcal{A})$ and $\mathbf{x}, \mathbf{y} \in \mathbf{2}^k$, is $\mathbf{x}{:}\mathbf{y} \in \mathbf{orb}(f)$.

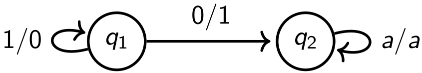
Orbit Complexity

The orbit problem is clearly decidable: the orbit of a string $\mathbf{x}$ is finite (bounded above by $2^{|\mathbf{x}|}$).

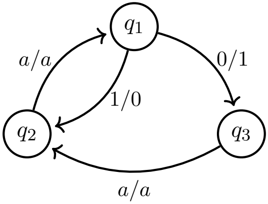
Orbit Complexity

The orbit problem is clearly decidable: the orbit of a string $\mathbf{x}$ is finite (bounded above by $2^{|\mathbf{x}|}$).

Also, this bound is tight. The adding machine achieves it.



Poll



For state q_1 and for $\mathbf{x}, \mathbf{y} \in \mathbf{2}^{1,000,000}$, how quickly can we solve the orbit problem?

- seconds?
- minutes?
- days?
- years?
- longer?

Rationality

Goal: Determine when $\mathbf{orb}(f)$ is regular.

Rationality

Goal: Determine when $\mathbf{orb}(f)$ is regular.
Brzozowski: it suffices to show $\mathbf{orb}(f)$ has finitely many quotients.

Rationality

Goal: Determine when $\mathbf{orb}(f)$ is regular.
Brzozowski: it suffices to show $\mathbf{orb}(f)$ has finitely many quotients.
We can describe the quotients by adding a translation function. Let

$$\mathbf{R}(f,g) = \{\mathbf{x} : \mathbf{y} \mid \exists t \in \mathbb{Z} \text{ such that } g(f^t(\mathbf{x})) = \mathbf{y}\}$$

Then $\mathbf{R}$ is closed under quotients: if $f, g \in \mathcal{G}(\mathcal{A})$ and $\mathbf{b} = g(\mathbf{a})$, then

$$(\mathbf{a}{:}\mathbf{b})^{-1}\mathbf{R}(f,g) = \begin{cases} \mathbf{R}(f_\mathbf{a}, g_\mathbf{a}) & \text{if } f \text{ is even,} \\ \mathbf{R}({f^2}_\mathbf{a}, g_\mathbf{a}) & \text{if } f \text{ is odd.} \end{cases}$$

$$(\mathbf{a}{:}\overline{\mathbf{b}})^{-1}\mathbf{R}(f,g) = \begin{cases} \emptyset & \text{if } f \text{ is even,} \\ \mathbf{R}({f^2}_\mathbf{a}, f_\mathbf{a} g_{\overline{\mathbf{a}}}) & \text{if } f \text{ is odd.} \end{cases}$$

Abelian Orbits

In abelian automaton groups, there are finitely many quotients if and only if there are finitely many first components (because $\mathcal{G}(\mathcal{A})$ is "contracting").

Abelian Orbits

In abelian automaton groups, there are finitely many quotients if and only if there are finitely many first components (because $\mathcal{G}(\mathcal{A})$ is "contracting").

Further, the first components are determined by the simple map

$$\varphi(f) = \begin{cases} f_{\mathbf{a}} & \text{if } f \text{ is even,} \\ {f^2}_{\mathbf{a}} & \text{if } f \text{ is odd.} \end{cases}$$

Thus we must only determine if $\varphi^*(f)$ is finite.

Algebra to the Rescue

We can associate to $\mathcal{G}(\mathcal{A})$ an algebraic number α and an embedding $\Psi : \mathcal{G}(\mathcal{A}) \to \mathbb{Q}(\alpha)$ such that if f is even, then $\Psi(f_{\mathbf{a}}) = \alpha\Psi(f)$.

Algebra to the Rescue

We can associate to $\mathcal{G}(\mathcal{A})$ an algebraic number α and an embedding $\Psi : \mathcal{G}(\mathcal{A}) \to \mathbb{Q}(\alpha)$ such that if f is even, then $\Psi(f_{\mathbf{a}}) = \alpha\Psi(f)$.

Under Ψ, φ has the simple form

$$\Psi(\varphi(f)) = \begin{cases} \alpha\Psi(f) & \text{if } f \text{ is even,} \\ 2\alpha\Psi(f) & \text{if } f \text{ is odd.} \end{cases}$$

Algebra to the Rescue

We can associate to $\mathcal{G}(\mathcal{A})$ an algebraic number α and an embedding $\Psi : \mathcal{G}(\mathcal{A}) \to \mathbb{Q}(\alpha)$ such that if f is even, then $\Psi(f_{\mathsf{a}}) = \alpha\Psi(f)$.

Under Ψ, φ has the simple form

$$\Psi(\varphi(f)) = \begin{cases} \alpha\Psi(f) & \text{if } f \text{ is even,} \\ 2\alpha\Psi(f) & \text{if } f \text{ is odd.} \end{cases}$$

Using a few more tricks, the following theorem follows:

Theorem: $\mathbf{orb}(f)$ is rational if and only if some power of α is rational.

Table of Contents

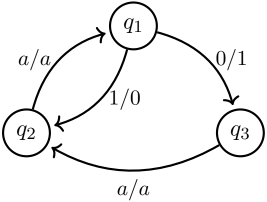
Goal

Embed $\mathcal{G}(\mathcal{A})$ in some algebraic object, while preserving residuation structure in a useful way.

System of Equations

Want to find a map $\Psi : \mathcal{G}(\mathcal{A}) \to \mathbb{Q}(\alpha)$ where residuation is given by

$$\Psi(f_{\mathbf{a}}) = \begin{cases} \alpha\Psi(f) & \text{if } f \text{ is even,} \\ \alpha\Psi(f) + (-1)^a & \text{if } f \text{ is odd.} \end{cases}$$



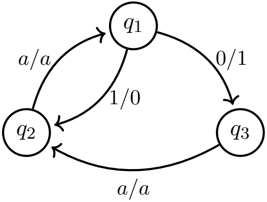
$\Longrightarrow$

$$\begin{aligned} zx_1 + 1 &= x_3 \\ zx_1 - 1 &= x_2 \\ zx_3 &= x_2 \\ zx_2 &= x_1 \end{aligned}$$

System of Equations

Want to find a map $\Psi : \mathcal{G}(\mathcal{A}) \to \mathbb{Q}(\alpha)$ where residuation is given by

$$\Psi(f_{\mathsf{a}}) = \begin{cases} \alpha\Psi(f) & \text{if } f \text{ is even,} \\ \alpha\Psi(f) + (-1)^{a} & \text{if } f \text{ is odd.} \end{cases}$$



$$\Longrightarrow$$

$$\begin{aligned} zx_1 + 1 &= x_3 \\ zx_1 - 1 &= x_2 \\ zx_3 &= x_2 \\ zx_2 &= x_1 \end{aligned}$$

Solving this system equations will produce suitable values for α, $\Psi(q_1), \ldots, \Psi(q_n)$.

How to solve it

This system can interpreted as an ideal of a polynomial ring with unknowns for α and for each state in $\mathcal{A}$. Computing a Gröbner basis of this ideal gives an easy way to enumerate the solutions.

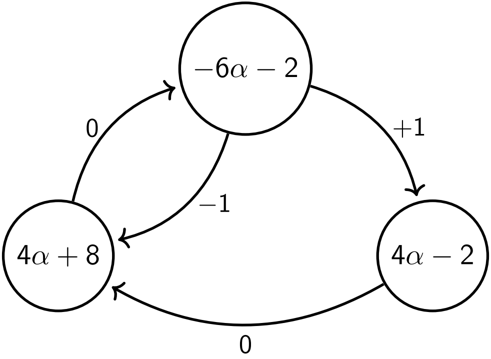
How to solve it

This system can interpreted as an ideal of a polynomial ring with unknowns for α and for each state in $\mathcal{A}$. Computing a Gröbner basis of this ideal gives an easy way to enumerate the solutions.

In general, Gröbner bases are expensive to compute (EXPSPACE-complete), but because our equations are "nearly linear", we can compute a Gröbner basis in time $O(n^6)$.

Solution

α has minimal polynomial $\chi(z) = z^2 + z + 1/2$.



(The embeddings shown are scaled by 5 for simplicity)

Summary

By computing an algebraic number associated to $\mathcal{G}(\mathcal{A})$, we can check if $\mathcal{A}$ has a rational orbit relation in time $O(n^6)$.

Summary

By computing an algebraic number associated to $\mathcal{G}(\mathcal{A})$, we can check if $\mathcal{A}$ has a rational orbit relation in time $O(n^6)$.

We expect this algebraic tool to assist in other computational problems arising in these automata.

Summary

By computing an algebraic number associated to $\mathcal{G}(\mathcal{A})$, we can check if $\mathcal{A}$ has a rational orbit relation in time $O(n^6)$.

We expect this algebraic tool to assist in other computational problems arising in these automata.

All algorithms discussed were implemented in Sage; code is available at

`https://github.com/tim-becker/thesis-code`.

Acknowledgements

Klaus Sutner for advising this project.

Evan Bergeron for many helpful conversations.

Questions?