Mealy machines, automaton (semi)groups, decision problems, and random generation

Thibault Godin
PhD defense, July 13, 2017
Invertible reversible non-coreversible automata generate infinite non-Burnside groups [LATA’15 w. Klimann and Picantin]

Infinite Burnside

Finite groups

Random generation

Finiteness

Infinite groups

Mealy automata

Growth

Automaton patterns and group properties

Dynamics of the action

Singular points

Schreier graphs

Wang tilings
Mealy automata

random generation

finite groups

infinite groups

dynamics of the action

Schreier graphs

Wang tills

singular points

automaton patterns and group properties

finiteness in infinite Burnside growth

c\ a\ b\ d\ e

\|0

0|0

1|1

0|0

1|0

0|0

1|1

\|0

\|1

\|0

\|1

\|0

\|1

Mealy automata

Analogue to Dixon theorem [ANALCO'16]

Bireversible automata of prime size cannot generate infinite Burnside groups [MFCS'16 & ToCS'17 w. Klimann]

Level transitive reversible automata have exponential growth [Klimann'16]

Invertible reversible non-coreversible automata generate infinite non-Burnside groups [LATA'15 w. Klimann and Picantin]

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR'16]
Mealy automaton $\mathcal{G}$

$A = (Q, \Sigma, \delta, \rho)$
Mealy automaton $G$

$A = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma \rightarrow \Sigma , \ q \in Q$

$d \rightarrow b$

$\rho_{d}(10001) = \rho_{a}(\rho_{d}(10001))$
$G = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \rightarrow \Sigma^*, \ q \in Q$

$\langle A \rangle := \langle \rho_q | q \in Q \rangle$

$\rho_d(10001) = \rho_a(\rho_d(10001))$

$M = \sum_{i=0}^{\infty} \lambda_i d_i$

Mealy automaton $G$
A = (Q, Σ, δ, ρ)

ρ_q : Σ^* → Σ^*, q ∈ Q

Mealy automaton G
Mealy automaton $\mathcal{G}$

$\mathcal{A} = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \to \Sigma^*, \ q \in Q$

$$d \xrightarrow{1} b \xrightarrow{0} a$$
Mealy automaton $\mathcal{G}$

$\mathcal{A} = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \to \Sigma^*, \ q \in Q$

$\rho_d(10001) = \rho_a(\rho_d(10001))$

$\langle A \rangle := \langle \rho_q \mid q \in Q \rangle$
Mealy automaton $G$

$A = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \rightarrow \Sigma^*$, $q \in Q$

$\rho_d(10001) = \rho_a(\rho_d(10001))$

$\langle A \rangle := \langle \rho_q \mid q \in Q \rangle$
A = (Q, Σ, δ, ρ)
ρ_q : Σ^* → Σ^*, q ∈ Q

Mealy automaton \( G \)
Mealy automaton $\mathcal{G}$

$A = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \to \Sigma^*$, $q \in Q^*$

$\rho_d : \{10001\} \to \{10001\}$
Mealy automaton $\mathcal{G}$

$\mathcal{A} = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \rightarrow \Sigma^*$, $q \in Q^*$
Mealy automaton $\mathcal{G}$

$\mathcal{A} = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \to \Sigma^*, \; q \in Q^*$

$\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
\end{array}$
Mealy automaton $\mathcal{G}$

$\mathcal{A} = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \rightarrow \Sigma^*, \ q \in Q^*$

$\rho_{da}(10001) = \rho_a(\rho_d(10001))$
Mealy automaton $\mathcal{G}$

$\mathcal{A} = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \to \Sigma^*$, $q \in Q^*$

$\rho_{da}(10001) = \rho_a(\rho_d(10001))$

$\langle \mathcal{A} \rangle := \langle \rho_q \mid q \in Q^* \rangle$
Mealy automaton $G$

$A = (Q, \Sigma, \delta, \rho)$

$\rho_q : \Sigma^* \rightarrow \Sigma^*$, $q \in Q^*$

$\rho_{da}(10001) = \rho_a(\rho_d(10001))$

$\langle A \rangle := \langle \rho_q \mid q \in Q^* \rangle$

$da$ is a state of $G^2$
Order

Order of an element

$x \in G$ has finite order if $\exists n \geq 1, x^n = e$
Order

Order of an element

$x \in G$ has finite order if $\exists n \geq 1, x^n = e$

- $\mathbb{Z}/n\mathbb{Z}$: every element has finite order
- $\mathbb{Z}$: 0 is the only element of finite order
- On the circle $\mathbb{R}/2\pi\mathbb{Z}$: $\pi/2$ has finite order, but 1 has infinite order
The Burnside problem

Burnside (1902):
Can a finitely generated group have all elements of finite order and be infinite?

Golod and Shafarevich: yes! (1964)
Aleshin+Grigorchuk: an example generated by a Mealy automaton (1972+1980)
The Burnside problem

**Burnside (1902):**
Can a finitely generated group have all elements of finite order and be infinite?

**Golod and Shafarevich: yes! (1964)**
The Burnside problem

**Burnside (1902):**
Can a finitely generated group have all elements of finite order and be infinite?

**Golod and Shafarevich: yes! (1964)**

**Aleshin+Grigorchuk:**
an example generated by a Mealy automaton (1972+1980)
The Burnside problem

**Burnside (1902):**
Can a finitely generated group have all elements of finite order and be infinite?

**Golod and Shafarevich: yes! (1964)**

**Aleshin+Grigorchuk:** an example generated by a Mealy automaton (1972+1980)
Mealy automata

Infinite Burnside patterns and group properties

Finiteness

Random generation

Finite groups

Infinite groups

Dynamics of the action

Singular points

Schreier graphs

Wang tilings

Invertible reversible non-coreversible automata generate infinite non Burnside groups

[LATA’15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups

[MFCS’16 w. Klimann]

Analogue to Dixon theorem

[ANALCO’16]

The set of singular points of a contracting automaton is described by a Büchi automaton

[DGKPR’16]

Level transitive reversible automata have exponential growth

[Klimann’16]

Invertible reversible non-coreversible automata generate infinite non Burnside groups

[LATA’15 w. Klimann and Picantin]

Invertible reversible non-coreversible automata generate infinite non Burnside groups

[LATA’15 w. Klimann and Picantin]
Invertible reversible non-coreversible automata generate infinite non Burnside groups
[LATA’15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups
[MFCS’16 w. Klimann]

The set of singular points of a contracting automaton is described by a Büchi automaton
[DGKPR’16]

Level transitive reversible automata have exponential growth
[Klimann’16]
Invertible reversible non-coreversible automata generate infinite non-Burnside groups [LATA'15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups [MFCS'16 w. Klimann]

Analogue to Dixon theorem [ANALCO'16]

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR'16]

Level transitive reversible automata have exponential growth [Klimann'16] [LATA'15 w. Klimann and Picantin] [MFCS'16 & ToCS'17 w. Klimann] [ANALCO'16] [arXiv'16] w. D'Angeli, Klimann, Picantin, and Rodaro
Analogue to Dixon theorem [ANALCO'16]

\[ \langle \sigma \rangle = \begin{cases} \mathcal{S}_k \times \mathcal{S}_k \\
(A_k \times A_k) \rtimes \langle (\pi, \pi) \rangle \\
A_k \times A_k \end{cases} \]
Finite random groups

Theorem

Any finite group $G$ is a subgroup of $\mathfrak{S}_{|G|}$.  

Theorem (Dixon, 1969)
Finite random groups

Theorem
Any finite group $G$ is a subgroup of $\mathfrak{S}_{|G|}$.

First idea
Pick up some permutations $\sigma_1, \ldots, \sigma_n$ of $\{1, \ldots, k\}$, look at $\langle \sigma_1, \ldots, \sigma_n \rangle$. 

Theorem (Dixon, 1969) w.g.p. $\langle \sigma, \tau \rangle = \mathfrak{S}_k \mathfrak{A}_k \mathfrak{S}_k$ # permutations cyclic groups?
Finite random groups

**Theorem**
Any finite group $G$ is a subgroup of $\mathfrak{S}_{|G|}$.

**First idea**
Pick up some permutations $\sigma_1, \ldots, \sigma_n$ of $\{1, \ldots, k\}$, look at $\langle \sigma_1, \ldots, \sigma_n \rangle$. 

\begin{align*}
1 & \quad 2 & \quad 3 & \quad \ldots & \quad k! \\
\text{# permutations}
\end{align*}
Finite random groups

Theorem
Any finite group $G$ is a subgroup of $S_{|G|}$.

First idea
Pick up some permutations $\sigma_1, \ldots, \sigma_n$ of $\{1, \ldots, k\}$, look at $\langle \sigma_1, \ldots, \sigma_n \rangle$.
**Finite random groups**

**Theorem**
Any finite group $G$ is a subgroup of $\mathfrak{S}_{|G|}$.

**First idea**
Pick up some permutations $\sigma_1, \ldots, \sigma_n$ of $\{1, \ldots, k\}$, look at $\langle \sigma_1, \ldots, \sigma_n \rangle$.

![Diagram](attachment:diagram.png)

- 1, 2, 3, ..., $k$!
- # permutations
- $\mathfrak{S}_k$
- cyclic groups
**Finite random groups**

**Theorem**
Any finite group $G$ is a subgroup of $\mathfrak{S}_{|G|}$.

**First idea**
Pick up some permutations $\sigma_1, \ldots, \sigma_n$ of $\{1, \ldots, k\}$, look at $\langle \sigma_1, \ldots, \sigma_n \rangle$. 

```
1 2 3 ...
k! # permutations
```

**cyclic groups**

$\mathfrak{S}_k$
Theorem (Dixon, 1969)

\[ \text{w.g.p. } \langle \sigma, \tau \rangle = \begin{cases} \mathcal{S}_k \\ \mathcal{A}_k \end{cases} \]

finite random groups

1 2 3 \[ k! \] # permutations
cyclic groups
Finite random groups

Theorem (Dixon, 1969)

w.g.p. \( \langle \sigma, \tau \rangle = \begin{cases} \mathcal{S}_k \\ \mathcal{A}_k \end{cases} \)

\( 1 \) \hspace{1cm} \( 2 \) \hspace{1cm} \( 3 \) \hspace{1cm} \( k! \) \hspace{1cm} \# \text{ permutations}

cyclic groups

\( \mathcal{S}_k \)

\( \mathcal{A}_k \)
Finite random groups

Theorem (Dixon, 1969)

\[
w.g.p. \langle \sigma, \tau \rangle = \begin{cases} \mathfrak{S}_k \\ \mathfrak{A}_k \end{cases}
\]

\(\mathfrak{S}_k\) or \(\mathfrak{A}_k\)

cyclic groups

1 2 3

\(k!\) # permutations
Random automata

Is the generated group finite?
Random automata

Is the generated group finite?

Yes, size $2^{64} \cdot 3^4$. 

Antonenko + Russeiev
Random automata

Is the generated group finite?

Yes, size $2^{64} \cdot 3^4$.

Difficult problem + inefficient rejection sampling.
Random automata
Random automata

Antonenko + Russeiev

cyclic automata
Random 2-state cyclic automata

\[
\langle \sigma, \tau \rangle = \langle (\sigma, \tau), (\tau, \sigma) \rangle
\]
Random 2-state cyclic automata

\[
\langle \begin{array}{c}
\circlearrowright \\
\circlearrowleft
\end{array}\rangle = \langle (\sigma, \tau), (\tau, \sigma) \rangle
\]

Contribution

\[
\langle \begin{array}{c}
\circlearrowright \\
\circlearrowleft
\end{array}\rangle = \begin{cases} \mathcal{S}_k \times \mathcal{S}_k \\ (\mathcal{A}_k \times \mathcal{A}_k) \times \langle (\pi, \pi) \rangle \\ \mathcal{A}_k \times \mathcal{A}_k \end{cases}
\]

\[
\mathcal{A}_k \times \mathcal{S}_k
\]

\[
\mathcal{S}_k \times \mathcal{A}_k
\]
Random 2-state cyclic automata

\[ \langle \sigma \tau \rangle = \langle (\sigma, \tau), (\tau, \sigma) \rangle \]

Contribution

\[ \langle \sigma \tau \rangle = \begin{cases} S_k \times S_k \\ (A_k \times A_k) \rtimes \langle (\pi, \pi) \rangle \\ A_k \times A_k \end{cases} \]
Random 2-state cyclic automata

\[
\langle \sigma \tau \rangle = \langle (\sigma, \tau), (\tau, \sigma) \rangle
\]

Contribution

\[
\langle \sigma \tau \rangle = \begin{cases} 
S_k \times S_k \\
(A_k \times A_k) \times \langle (\pi, \pi) \rangle \\
A_k \times A_k 
\end{cases}
\]
Random automata

Antonenko + Russeiev

cyclic automata
Dixon like

structurally finite

\[ \sigma_1, \ldots, \sigma_n \]

Dixon like (conj.)

\[ a \]

\[ b \]

\[ c \]

\[ d \]

\[ e \]

\[ f \]

\[ \sigma_1 \]

\[ 1 \]

\[ \sigma_n \]

\[ n \]

\[ \tau \]

\[ \Theta_k \]

\[ a_n \]

\[ \Theta_k \times \Theta_k \]

\[ a_n \times a_n \times (r, s) \]

structurally infinite

finite by construction

\[ \mod \] reduction

decidable finiteness

2-state bireversible automata

"complexity"
Mealy automata

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR'16]

The dynamics of the action

Wang tillings

Schreier graphs

finite groups

infinite groups

dynamics of the action

singular points

Invertible reversible non-coreversible automata generate infinite non-Burnside groups [LATA'15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups [MFCS'16 w. Klimann]

Analogue to Dixon theorem [ANALCO'16]

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR'16]

Stabilisers and singular points

The stabilisers of an infinite point $\xi$ is $\text{Stab}_{\langle A \rangle}(\xi) = \{g \in \langle A \rangle \mid g(\xi) = \xi\}$
Stabilisers and singular points

The stabilisers of an infinite point $\xi$ is $\text{Stab}_{\langle A \rangle}(\xi) = \{ g \in \langle A \rangle \mid g(\xi) = \xi \}$
Stabilisers and singular points

The stabilisers of an infinite point $\xi$ is $\text{Stab}_\langle \mathcal{A} \rangle (\xi) = \{ g \in \langle \mathcal{A} \rangle \mid g(\xi) = \xi \}$

Example

$\rho_e, \rho_b, \rho_c, \rho_d \in \text{Stab}_\langle \mathcal{G} \rangle (1^\omega)$

studied by Y. Vorobets
Stabilisers and singular points

The stabilisers of an infinite point $\xi$ is $\text{Stab}_{\langle A \rangle}(\xi) = \{ g \in \langle A \rangle \mid g(\xi) = \xi \}$

Example

$\rho_e, \rho_b, \rho_c, \rho_d \in \text{Stab}_{\langle G \rangle}(1^\omega)$

studied by Y. Vorobets

Interesting elements

$2^\omega$ is stabilised by $\rho_a$
Stabilisers and singular points

The stabilisers of an infinite point \( \xi \) is \( \text{Stab}_{\langle A \rangle}(\xi) = \{ g \in \langle A \rangle \mid g(\xi) = \xi \} \)

\[
\begin{array}{cccccc}
q & \circ & \circ & \circ & \circ & \cdots \\
\end{array}
\]

Example

\[
\begin{array}{cccccc}
 b & 1|1 & d \\
 0|0 & 1|1 & 0|0 \\
a & c & e \\
 0|1 & 1|0 & 0|0 \\
 1|1 & 1|0 & 1|1 \\
\end{array}
\]

\( \rho_e, \rho_b, \rho_c, \rho_d \in \text{Stab}_{\langle g \rangle}(1^\omega) \)

studied by Y. Vorobets

Interesting elements

\[
\begin{array}{ccc}
0|0 & 0|1 \\
1|1 & 1|0 \\
2|2 & 2|2 \\
\end{array}
\]

\( 2^\omega \) is stabilised by \( \rho_a \)

Singular points

\( \xi \) singular if \( \exists g \) stabilizing \( \xi \) and avoiding ending in \( e \)
Contracting automata

\[ \mathcal{A} \text{ contracting } \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in \mathcal{N} \]
Contracting automata

\( \mathcal{A} \) contracting \( \iff \exists \) finite \( \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in \mathcal{N} \)
Contracting automata

\[ \mathcal{A} \text{ contracting } \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in \mathcal{N} \]
Contracting automata

\[ A \text{ contracting } \iff \exists \text{ finite } N, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in N \]
Contracting automata

\[ \mathcal{A} \text{ contracting } \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in \mathcal{N} \]
Contracting automata

\( A \) contracting ⇐⇒ ∃ finite \( \mathcal{N} \), ∀ \( q, \xi \), ∃ \( n \), \( \delta_{\xi[:n]}(q) \in \mathcal{N} \)
Contracting automata

\[ \mathcal{A} \text{ contracting} \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in \mathcal{N} \]
Contracting automata

\[ \mathcal{A} \text{ contracting } \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[:n]}(q) \in \mathcal{N} \]
Contracting automata

\[ A \text{ contracting} \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[n]}(q) \in \mathcal{N} \]
Contracting automata

\[ A \text{ contracting} \iff \exists \text{ finite } \mathcal{N}, \forall q, \forall \xi, \exists n, \delta_{\xi[n]}(q) \in \mathcal{N} \]
Contracting automata and singular points

Basilica automaton

Büchi automaton

\[ \text{Contribution} \quad \text{Sing}(B) = \emptyset. \]
Contracting automata and singular points

Basilica automaton

N

\[
\begin{array}{ccc}
0 & 0 & 0 \\
\mathit{a} & \mathit{b} & \mathit{a} \\
\end{array}
\]

\[
\begin{array}{ccc}
0 & 1 & 0 \\
1 & 0 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{ccc}
\mathit{ba}^{-1} & \mathit{ab}^{-1} & \\
\mathit{a}^{-1} & \mathit{b}^{-1} & \\
\end{array}
\]
Contracting automata and singular points

\[
\begin{align*}
0 & \rightarrow b \\
0 & \\
 b & \rightarrow a \\
1 & \\
 a & \rightarrow e \\
\end{align*}
\]

Basilica automaton

\[
\begin{align*}
0|0 & \\
0|1 & \\
1|0 & \\
1|1 & \\
1|1,0|0 &
\end{align*}
\]

Büchi automaton

\[
\begin{align*}
0|1 & \\
1|0 & \\
0|1 & \\
1|0 & \\
0|0 &
\end{align*}
\]
Contracting automata and singular points

\[ \xi \]

\[ A_m^q \]

\[ \xi[0] \]

\[ \xi[1] \]

\[ \xi[\ell] \]

\[ \xi[\ell+1] \]

\[ N \]

\[ a^0 \]

\[ b^1 \]

\[ a \rightarrow b \rightarrow a \in \mathcal{N} \]

\[ b \rightarrow a \rightarrow e \]

\[ 1 \]

\[ a \rightarrow e \rightarrow e \]

Basilica automaton

\[ N \]

\[ ba^{-1} \]

\[ ab^{-1} \]

\[ e \]

\[ a^{-1} \]

\[ b^{-1} \]
Contracting automata and singular points

\[ \xi \text{ singular} \]

\[ q \xrightarrow{\xi[0]} \bigcirc \xrightarrow{\xi[1]} \bigcirc \xrightarrow{\xi[\ell]} \bigcirc \xrightarrow{\xi[\ell + 1]} \bigcirc \xrightarrow{\xi[\ell + 1]} \]

Basilica automaton

\[ 0|1 \quad 1|0 \quad 0|1 \quad 1|0 \quad 0|1 \quad 0|0 \]

\[ b \quad a \quad e \quad a^{-1} \quad b^{-1} \]

\[ \mathcal{N} \]

\[ 0|0 \quad 0|1 \quad 1|1 \quad 1|0 \quad 1|1, 0|0 \]
Contracting automata and singular points

\( A^m \)

\( \xi \) singular

\( \xi[0] \xrightarrow{q} \xi[1] \xrightarrow{} \xi[\ell] \xrightarrow{} \xi[\ell + 1] \)

Basilica automaton

\( N \)

\( b, a \rightarrow b^{-1}, a b^{-1} \)
Contracting automata and singular points

\[ \xi \text{ singular} \]

\[ A^m \]

\[ \xi[0] \rightarrow \circ \rightarrow \circ \rightarrow \circ \]

\[ \xi[\ell] \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \]

\[ \xi[0] \rightarrow \xi[1] \]

\[ \xi[\ell] \rightarrow \xi[\ell + 1] \]

Contribution

\[ \text{Sing}(B) = \emptyset. \]

Basilica automaton

\[ ba^{-1} \]

\[ ab^{-1} \]

\[ a^{-1} \]

\[ b^{-1} \]

\[ e \]

\[ \xi \text{ singular} \]

\[ \xi[0] \rightarrow \xi[1] \]
Contracting automata and singular points

\[ \xi[0] \xrightarrow{q} \xi[1] \]

\[ A^m \]

\[ \xi \text{ singular} \]

\[ \xi[0] \xrightarrow{\xi[0]} \xi[1] \]

\[ N \]

Basilica automaton

\[ ba^{-1} \]

\[ ab^{-1} \]

\[ b^{-1} \]

\[ a^{-1} \]

\[ e \]

\[ \xi[\ell] \xrightarrow{\xi[\ell]} \xi[\ell + 1] \]

\[ \xi[0] \xrightarrow{\xi[0]} \xi[1] \]

\[ \xi[\ell] \xrightarrow{\xi[\ell]} \xi[\ell + 1] \]

\[ \xi[\ell] \xrightarrow{\xi[\ell]} \xi[\ell + 1] \]

\[ 0|1 \]

\[ 1|0 \]

\[ 0|1 \]

\[ 1|1 \]

\[ 0|0 \]

\[ 0|1 \]

\[ 1|1 \]

\[ 1|0 \]

\[ 0|0 \]
Contracting automata and singular points

$A^m$

$\xi$ singular

$\xi[0] \rightarrow \circ \rightarrow \circ \rightarrow \circ$

$\xi[0] \rightarrow \circ \rightarrow \circ \rightarrow \circ$

$\xi[\ell] \rightarrow \circ \rightarrow \circ \rightarrow \circ$

Basilica automaton

$ba^{-1}$

$ab^{-1}$

$N$

$N$

$1|1, 0|0$
Contracting automata and singular points

Let $A^m$ be a $\xi$-singular automaton.

\[ \xi[0] \xrightarrow{\ell} \xi[1] \]

Büchi automaton

Let $\mathcal{N}$ be the Büchi automaton.

\[ \xi[\ell] \xrightarrow{\ell + 1} \]

The contribution of $\mathcal{N}$ is $\text{Sing}(\mathcal{B}) = \emptyset$.
Contracting automata and singular points

\[ \xi_{\text{singular}} \]

\[ A^m \]

Contribution

\[ \text{Sing}(\mathcal{B}) = \emptyset. \]
Contracting automata and singular points

\[ \xi_{[0]} \xrightarrow{q} \xi_{[1]} \]

\[ \xi_{[\ell]} \xrightarrow{N} \xi_{[\ell + 1]} \]

Proposition [Vorobets, DGKPR]

\[ \text{Sing}(G) = (0 + 1)^* \omega. \]
Contracting automata and singular points

\[ \xi[0] \rightarrow \xi[1] \]

\( q \)

\[ \xi[0] \rightarrow \xi[1] \]

\( \xi[0] \rightarrow \xi[1] \)

\[ \xi[\ell] \rightarrow \xi[\ell + 1] \]

\[ \xi[\ell] \rightarrow \xi[\ell + 1] \]

\[ N \]

Proposition [Vorobets, DGKPR]

\( \text{Sing}(G) = (0 + 1)^* \omega \).
Contracting automata and singular points

\[ A^m \]

\[ \xi \text{ singular} \]

Proposition [Vorobets, DGKPR]

\[ \text{Sing}(\mathcal{G}) = (0 + 1)^*1^\omega. \]
Invertible reversible non-coreversible automata generate infinite non-Burnside groups [LATA’15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups [MFCS’16 w. Klimann]

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR’16]

Infinite Burnside

Finiteness

Automaton patterns and group properties

Growth

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR’16]
About the Grigorchuk automaton

Reversibility:
Each input letter permutes the stateset.

Actions of the states on the letters:
\( \rho_a: 0 \mapsto 1, 1 \mapsto 0 \)
\( \rho_b, \rho_c, \rho_d, \rho_e: 0 \mapsto 0, 1 \mapsto 1 \)

Action of a letter on the states:
\( \delta_0: a, d, e \mapsto e; b, c \mapsto a \)

\[ \rightarrow \text{permutations} \]
\[ \rightarrow \text{invertible} \]

\[ \rightarrow \text{non-reversible} \]
About the Grigorchuk automaton

Actions of the states on the letters:

\[ \rho_a : 0 \mapsto 1 \mapsto 0 \]
\[ \rho_b, \rho_c, \rho_d, \rho_e : 0 \mapsto 0; \quad 1 \mapsto 1 \]

→ permutations
About the Grigorchuk automaton

Actions of the states on the letters:

\[ \rho_a : 0 \mapsto 1 \mapsto 0 \]
\[ \rho_b, \rho_c, \rho_d, \rho_e : 0 \mapsto 0; \quad 1 \mapsto 1 \]

→ permutations
→ invertible
About the Grigorchuk automaton

Actions of the states on the letters:

- $\rho_a : 0 \mapsto 1 \mapsto 0$
- $\rho_b, \rho_c, \rho_d, \rho_e : 0 \mapsto 0; \quad 1 \mapsto 1$

→ permutations
→ invertible

Action of a letter on the states:

- $\delta_0 : a, d, e \mapsto e; \quad b, c \mapsto a$

→ not a permutation
About the Grigorchuk automaton

Actions of the states on the letters:

\[ \rho_a : 0 \rightarrow 1 \rightarrow 0 \]
\[ \rho_b, \rho_c, \rho_d, \rho_e : 0 \rightarrow 0; \quad 1 \rightarrow 1 \]

→ permutations
→ invertible

Action of a letter on the states:

\[ \delta_0 : a, d, e \rightarrow e; \quad b, c \rightarrow a \]

→ not a permutation
→ non-reversible
About the Grigorchuk automaton

Reversibility:
Each input letter permutes the stateset.

Actions of the states on the letters:
\( \rho_a : 0 \mapsfrom 1 \mapsfrom 0 \)
\( \rho_b, \rho_c, \rho_d, \rho_e : 0 \mapsfrom 0; \ 1 \mapsfrom 1 \)

→ permutations
→ invertible

Action of a letter on the states:
\( \delta_0 : a, d, e \mapsfrom e; \ b, c \mapsfrom a \)

→ not a permutation
→ non-reversible
About the Grigorchuk automaton

Reversibility:
Each input letter permutes the stateset.

Actions of the states on the letters:
\[
\begin{align*}
\rho_a &: 0 \mapsto 1 \mapsto 0 \\
\rho_b, \rho_c, \rho_d, \rho_e &: 0 \mapsto 0; \quad 1 \mapsto 1
\end{align*}
\]

→ permutations
→ invertible

Action of a letter on the states:
\[
\delta_0 : a, d, e \mapsto e; \quad b, c \mapsto a
\]

→ not a permutation
→ non-reversible
**Observation**

Every known automaton generating an infinite Burnside group happens to be non-reversible.

**Question**

Can a reversible automaton generate an infinite Burnside group?
Theorem(s)

An invertible and reversible automata which is:

- 2-state
- connected 3-state
- non coreversible
- connected with prime size

[Klimann]

cannot generate an infinite Burnside group.
Theorem(s)

An invertible and reversible automata which is:

2-state

[Klimann]

STACS'13

cannot generate an infinite Burnside group.
Theorem(s)

An invertible and reversible automata which is:

2-state connected 3-state

[Klimann] [Klimann, Picantin, and Savchuk]

STACS’13 DLT’15

cannot generate an infinite Burnside group.
Theorem(s)

An invertible and reversible automata which is:

- 2-state connected 3-state non coreversible

[Klimann] [Klimann, Picantin, and Savchuk] [G., Klimann, and Picantin]

STACS'13 DLT'15 LATA'15

cannot generate an infinite Burnside group.
**Theorem(s)**

An invertible and reversible automata which is:

- 2-state connected
- 3-state non-coreversible
- connected with prime size

[Klimann] [Klimann, Picantin, and Savchuk] [G., Klimann, and Picantin] [G. and Klimann]

STACS'13 DLT'15 LATA'15 MFCS'16

cannot generate an infinite Burnside group.
Schreier tree

\[ A \]
Schreier tree

\[ A \]

\[ A^2 \]
Schreier tree

$A$

$A^2$

22 / 30
Schreier tree

\[ A \]

\[ A^2 \]
Schreier tree

The connected component of $A^2$ containing $ab$

Size ratio

$A^0$

$A^1$

$A^2$

$A^3$
Boundedness

Proposition

\(\langle A \rangle\) is finite iff the labels of the cc of \((A^n)_n\) are ultimately 1.
Boundedness

Proposition
\langle A \rangle \text{ is finite iff the labels of the cc of } (A^n)_n \text{ are ultimately 1.}

Proposition
\rho_q \text{ has finite order iff the labels of the cc containing } q^n \text{ are ultimately 1.}
Proposition

\( e \) liftable to \( f \) \( \Rightarrow \) \( \text{label}(e) \leq \text{label}(f) \).
Liftable

Proposition

e liftable to $f \Rightarrow \text{label}(e) \leq \text{label}(f)$.
Liftable

**Proposition**

\[ e \text{ liftable to } f \Rightarrow \text{label}(e) \leq \text{label}(f). \]
Liftable

Proposition

e liftable to \( f \Rightarrow \text{label}(e) \leq \text{label}(f) \).
Liftable paths

```
Liftable paths

```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```
Liftable paths

```
6
  |
  a
  |
  2 4
 / \ /
|  | |
aa ab
|
| |
2 4
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
Liftable paths
Liftable paths

```
6
a

aa

aba

aaa

aab

aaba

aabc

aba

baba

abad

abae

abca

abc

abcb
```
Jungle tree
active ≡ labels not ending with $1^\omega$.
If active liftable path ✓: not Burnside.
Otherwise:
Jungle tree

active ≡ labels not ending with $1^\omega$.

If active liftable path ✓: not Burnside.

Otherwise:

```
    7
   /|
  2 /
 /|
a aa
|
1|
|
1  1
|
aaa aab
|
1  1
|
1  1
|
1  1
|
aaaa aaab aabc aabd
```
Jungle tree

active $\equiv$ labels not ending with $1^\omega$.

If active liftable path $\checkmark$: not Burnside.

Otherwise:

words in the jungle tree have bounded order.
3-state case

Every word in the tree ✓

Not every word in the tree

\[ A^{\ell-1} \]

\[ A^{\ell} \]

\[ A^{\ell+1} \]
3-state case
3-state case
3-state case

Every word in the tree

Not every word in the tree
3-state case

Every word in the tree

Not every word in the tree
3-state case

Every word in the tree

not every word in the tree
3-state case

Every word in the tree ✓

Jungle tree
3-state case

Every word in the tree ✓
Looking for (equivalent) words

Idea: ∀x₀x₁x₂⋯ find a word with the same action in the jungle tree

Only 1's

| 7 |

2

only 1's
Looking for (equivalent) words

Idea: $\forall x_0 x_1 x_2 \cdots$ find a word with same action in the jungle tree
Looking for (equivalent) words

Idea: $\forall x_0 x_1 x_2 \cdots$ find a word with same action in the jungle tree
Looking for (equivalent) words

Idea: $\forall x_0 x_1 x_2 \cdots$ find a word with same action in the jungle tree
Looking for (equivalent) words

Idea: $\forall x_0 x_1 x_2 \ldots$ find a word with same action in the jungle tree
Invertible reversible non-coreversible automata generate infinite non Burnside groups [LATA’15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups [MFCS’16 w. Klimann]

Infinite Burnside

Pattern and group properties

Finiteness

Growth

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR’16]
Invertible reversible non-coreversible automata generate infinite non-Burnside groups [LATA’15 w. Klimann and Picantin]

Bireversible automata of prime size cannot generate infinite Burnside groups [MFCS’16 w. Klimann]

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR’16]

Infinite Burnside

Infinite groups

Dynamics of the action

Schreier graphs

Wang tilings

analytic

Finiteness

automaton patterns and group properties

finite groups

infinite groups

singular points

automaton patterns and group properties

In invertible reversible automata, patterns and group properties relate to exponential growth [Klimann’16]

The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR’16]

Level transitive reversible automata have exponential growth [Klimann’16]
The set of singular points of a contracting automaton is described by a Büchi automaton [DGKPR'16].
\[
\langle \sigma, \tau \rangle = \begin{cases} 
S_k \times S_k \\
(A_k \times A_k) \rtimes \langle (\pi, \pi) \rangle \\
A_k \times A_k
\end{cases}
\]

Analogue to Dixon theorem [ANALCO'16]
Mealy automata

- Random generation
- Finite groups
- Infinite groups
- Dynamics of the action
- Schreier graphs
- Wang tilings
- Singular points
- Automaton patterns and group properties
- Finiteness of infinite Burnside growth
- \( c_a b d e \)

- Invertible reversible non-coreversible automata generate infinite non-Burnside groups
  - [LATA’15 w. Klimann and Picantin]
- Bireversible automata of prime size cannot generate infinite Burnside groups
  - [MFCS’16 w. Klimann]
- Analogue to Dixon theorem
  - [ANALCO’16]
- The set of singular points of a contracting automaton is described by a Büchi automaton
  - [DGKPR’16]
- Level transitive reversible automata have exponential growth
  - [Klimann’16]
- [LATA’15 w. Klimann and Picantin]
- [MFCS’16 & ToCS’17 w. Klimann]
- [ANALCO’16]
- [arXiv’16 w. D’Angeli, Klimann, Picantin, and Rodaro]