# Automata, languages, and groups of automorphisms of rooted trees

## Part II - Groups and Automata: a perfect match

Marialaura Noce

Georg-August-Universität Göttingen

## Plan for today

1. Previously, on “Automata, languages, and groups of automorphisms of rooted trees”
2. Introduction to Grammars
3. Part II: Groups and automata
4. Automata groups
5. Mealy machine
6. How to generate automata groups
7. Let’s start now with Part III: groups of automorphisms of rooted trees
8. Plan for Thursday

# Previously, on “Automata, languages, and groups of automorphisms of rooted trees”

# What we did yesterday

- Basic notions in Automata Theory
- Deterministic and non deterministic finite state automata
    - Differences
    - How one can construct a DFA from a NDFA and vice versa.
- Languages, and regular languages
- The pumping lemma

- We wanted to prove that some language is not regular. Take
  - $L = \{0^m 1^m \mid m \geq 0\}$.
  - $L = \{0^m \mid m \text{ is a prime number}\}$.

# Proof of I

Claim: The language $L = \{0^m 1^m \mid m \geq 0\}$ is not regular.

**Proof.**

$\square$

# Proof of II

Claim: The language $L = \{0^m \mid m \text{ is a prime number}\}$ is not regular.

**Proof.**

$\square$

# Introduction to Grammars

**From Wikipedia**

*Grammar* (again, from Ancient Greek [γραμματική]) is the set of structural rules governing the composition of clauses, phrases and words in a natural language.

**From Wikipedia**

*Grammar* (again, from Ancient Greek [γραμματική]) is the set of structural rules governing the composition of clauses, phrases and words in a natural language.

- Noam Chomsky gave a mathematical model of grammar in 1956 which is effective for writing computer languages.

# Grammar

Informally, a *grammar* can be seen like a deductive system, where the sentences of the generated language are its “theorems”.

# Formal definition

A grammar describes a way to generate a language by using some rules.
A grammar $G$ can be formally written as a 4-tuple $(T, N, S, P)$, where

- $T$ is a set of terminal symbols.
- $N$ is a set of variables or non-terminal symbols.
- $S$ is a variable from $N$ called the *start symbol*.
- $P$ is a set of production rules for terminals and non-terminals. A production rule has the form $\alpha \to \beta$, where $\alpha$ and $\beta$ are strings on $N \cup T$.

# Example

Let's consider a grammar to construct Italian sentences.

- $T$ = set of all Italian words

# Example

Let's consider a grammar to construct Italian sentences.

- $T$ = set of all Italian words
- $N$ = set of all non-terminals that correspond to the structural component in an Italian sentence (<sentence>, <subject>, <verb>, ...).

# Example

Let's consider a grammar to construct Italian sentences.

- $T$ = set of all Italian words
- $N$ = set of all non-terminals that correspond to the structural component in an Italian sentence (<sentence>, <subject>, <verb>, ...).
- Start symbol: <sentence>

# Example

Let’s consider a grammar to construct Italian sentences.

- $T$ = set of all Italian words
- $N$ = set of all non-terminals that correspond to the structural component in an Italian sentence (<sentence>, <subject>, <verb>, ...).
- Start symbol: <sentence>
- Production rules:
    - $< sentence > \rightarrow < subject >< verb >$
    - $< subject > \rightarrow < noun >$
    - $< verb > \rightarrow$ ascoltare
    - $< noun > \rightarrow$ pianoforte
    - $\ldots$ and many others if you want to learn Italian ;)

# Another example

You may recognize this example...
Let $G = (\{S\}, \{0, 1\}, S, P)$ with productions $P$ given by

- $S \to 0S1$
- $S \to 01$

# From grammars to languages

The set of all strings of terminal symbols that can be derived from a grammar by taking a start symbol and then consecutively applying production rules until reaching a string of terminal symbols is said to be the language generated from that grammar.

# From grammars to languages

The set of all strings of terminal symbols that can be derived from a grammar by taking a start symbol and then consecutively applying production rules until reaching a string of terminal symbols is said to be the language generated from that grammar. We write

$$L = \{w \in T^* \mid S \rightarrow_P w\}$$

The set of all strings of terminal symbols that can be derived from a grammar by taking a start symbol and then consecutively applying production rules until reaching a string of terminal symbols is said to be the language generated from that grammar. We write

$$L = \{w \in T^* \mid S \to_P w\}$$

**Example**

The grammar of the previous example can be used to describe the language $L = \{0^n 1^n \mid n \geq 0\}$.

# Example

Let $G = (\{0, 1, 2\}, \{S, A\}, S, P)$, where $P$ contains the following productions:

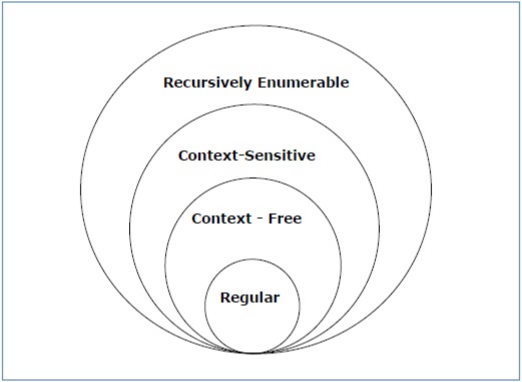
- $S \to 0SA2$
- $S \to \epsilon$
- $2A \to A2$
- $0A \to 01$
- $1A \to 11$

Which language is described by this grammar?

Noam Chomsky introduced four types of grammars.

| Type | Grammar | Language | Automaton |
|---|---|---|---|
| Type 0 | Unrestricted | Recursively enumerable | Turing Machine |
| Type 1 | Context sensitive | Context sensitive | Linear bounded |
| Type 2 | Context free | Context free | Pushdown |
| Type 3 | Regular | Regular | Finite state |

# Types of grammars II




Chomsky hierarchy.

# Regular grammars

A grammar $(N, T, S, P)$ is a regular grammar (Type 3) if one of the following hold:

- every production is of the form $A \to \beta B$ or $A \to \beta$ with $A, B \in N$ and $\beta \in T$ (right-linear grammar)
- every production is of the form $A \to B\beta$ or $A \to \beta$ with $A, B \in N$ and $\beta \in T$ (left-linear grammar)

# Regular grammars

A grammar $(N, T, S, P)$ is a regular grammar (Type 3) if one of the following hold:

- every production is of the form $A \to \beta B$ or $A \to \beta$ with $A, B \in N$ and $\beta \in T$ (right-linear grammar)
- every production is of the form $A \to B\beta$ or $A \to \beta$ with $A, B \in N$ and $\beta \in T$ (left-linear grammar)

**Example**

$G = (\{S, A\}, \{0, 1\}, S, P)$ with productions $P$ given by
$S \to 0S, S \to A, A \to 1A, A \to 1.$

A grammar $(N, T, S, P)$ is a regular grammar (Type 3) if one of the following hold:

- every production is of the form $A \to \beta B$ or $A \to \beta$ with $A, B \in N$ and $\beta \in T$ (right-linear grammar)
- every production is of the form $A \to B\beta$ or $A \to \beta$ with $A, B \in N$ and $\beta \in T$ (left-linear grammar)

**Example**

$G = (\{S, A\}, \{0, 1\}, S, P)$ with productions $P$ given by
$S \to 0S, S \to A, A \to 1A, A \to 1.$

**Theorem**

*A language is regular if and only if it is generated by a regular grammar.*

- Context-free grammars are grammars in which any of the production rules can be applied regardless of context (i.e. it does not depend on any other symbols that may or may not be around a given symbol that is having a rule applied to it).

- Context-free grammars are grammars in which any of the production rules can be applied regardless of context (i.e. it does not depend on any other symbols that may or may not be around a given symbol that is having a rule applied to it).
- Formally, a grammar is said to be *context-free* if its production rules are of this type:
  - $A \to \alpha$, where $A \in N$ and $\alpha \in (N \cup T)^*$.

# Context-free grammars and Pushdown automata

- Context-free grammars are grammars in which any of the production rules can be applied regardless of context (i.e. it does not depend on any other symbols that may or may not be around a given symbol that is having a rule applied to it).
- Formally, a grammar is said to be *context-free* if its production rules are of this type:
  - $A \to \alpha$, where $A \in N$ and $\alpha \in (N \cup T)^*$.

**Theorem**

*Context-free languages are generated by context-free grammars. The set of all context-free languages is the set of languages accepted by pushdown automata.*

- J.E. Hopcroft, R. Motwani, J. D. Ullman
  *Introduction to Automata Theory, Languages, and Computation*
  - Addison-Wesley Longman Publishing Co., Inc. 2006.
- R.W.Floyd, R. Beigel
  *The Language of Machines: an Introduction to Computability and Formal Languages*
  - Computer Science Press, New York, 1994.

# Part II: Groups and automata

- Let $X = \{x_1, x_2, \dots\}$, and $X^{-1} = \{x_1^{-1}, x_2^{-1}, \dots\}$, $X^{\pm} = X \cup X^{-1}$.
- The free reduction of a word $w$ on $X^{\pm}$ is obtained by replacing all subwords $x_i x_i^{-1}$ or $x_i^{-1} x_i$ from $w$ by the empty string $\varepsilon$ to form its free reduction[1].
  The resulting word is called the *free reduction of* $w$.

[1] It can be seen that the free reduction does not depend on the order of deletions.

# Words, relations, and group presentations

- Let $X = \{x_1, x_2, \ldots\}$, and $X^{-1} = \{x_1^{-1}, x_2^{-1}, \ldots\}$, $X^{\pm} = X \cup X^{-1}$.
- The free reduction of a word $w$ on $X^{\pm}$ is obtained by replacing all subwords $x_i x_i^{-1}$ or $x_i^{-1} x_i$ from $w$ by the empty string $\varepsilon$ to form its free reduction[1].
  The resulting word is called the *free reduction of* $w$.
- Given two words $w, w'$ on $X^{\pm}$, we write $w \sim w'$ to denote that the free reductions of $w$ and $w'$ are the same.
  For example: $abb^{-1}b \sim aa^{-1}ab^{-1}bb \sim ab$.

[1] It can be seen that the free reduction does not depend on the order of deletions.

# Words, relations, and group presentations II

We define the free group $F(X)$ by:

- the set of freely reduced words on $X^{\pm}$
- the multiplication of two elements $w_1, w_2 \in F(X)$ is the free reduction of the word $w_1 w_2$.
- The identity element in $F(X)$ is the empty string $\varepsilon$.

Note: there are multiple ways of defining a free group. This one suits our purposes.

- Given a set $R$ of words from $F(X)$, we let $\langle\langle R \rangle\rangle$ denote the subgroup of $F(X)$ generated by all conjugates of the elements from $R$.

# Group presentations

- Given a set $R$ of words from $F(X)$, we let $\langle\langle R\rangle\rangle$ denote the subgroup of $F(X)$ generated by all conjugates of the elements from $R$.
- We say that a group $G$ has the presentation $\langle X \mid R\rangle$ if it is isomorphic to the quotient group $F(X)/\langle\langle R\rangle\rangle$.
- We write $G = \langle X \mid R\rangle$. The elements from $R$ are called the *relations of the presentation*.
- If both $X$ and $R$ are finite then we say that $\langle X \mid R\rangle$ is a *finite presentation*.

# Group presentations

- Given a set $R$ of words from $F(X)$, we let $\langle\langle R\rangle\rangle$ denote the subgroup of $F(X)$ generated by all conjugates of the elements from $R$.
- We say that a group $G$ has the presentation $\langle X \mid R\rangle$ if it is isomorphic to the quotient group $F(X)/\langle\langle R\rangle\rangle$.
- We write $G = \langle X \mid R\rangle$. The elements from $R$ are called the *relations of the presentation*.
- If both $X$ and $R$ are finite then we say that $\langle X \mid R\rangle$ is a *finite presentation*.
- Given a word $w \in F(X)$, we write $w =_G 1$ if $\pi(w) = 1$, where $\pi : F(X) \to F(X)/\langle\langle R\rangle\rangle$ is the canonical projection from $F(X)$ to $G$.
- Informally, we can think of a word $w$ from $F(X)$ as representing an element of $G$, namely $\pi(w)$.

Informally, the presentation $\langle X \mid R \rangle$ indicates that we can take words from $F(X)$ and delete or insert subwords from $\langle\langle R \rangle\rangle$ without changing the element the word represents in $G$.

The area of group theory that studies groups via their presentations is called *combinatorial group theory*.

# Word problem

Informally, the presentation $\langle X \mid R \rangle$ indicates that we can take words from $F(X)$ and delete or insert subwords from $\langle\langle R \rangle\rangle$ without changing the element the word represents in $G$.

The area of group theory that studies groups via their presentations is called *combinatorial group theory*.

**Word problem for $G$**

The world problem for $G$ asks whether or not a given input word $w \in F(X)$ represents the identity in $G$, i.e. if $w =_G 1$.

We say that the word problem is *soluble* (or *decidable*) for $G$ if there exists a terminating algorithm that can decide on any input word $w$ whether $w =_G 1$.

# Example of groups with decidable word problem

- Automatic groups (such as finite groups, Braid groups, ...)
- Finitely generated free groups
- Finitely generated free abelian groups
- Polycyclic groups
- Finitely presented simple groups.
- Finitely presented residually finite groups
- One relator groups

# A group with unsolvable word problem

This example was given in 1986 by Collins. The group $G$ is generated by $X = \{a, b, d, c, e, p, q, r, t, k\}$, and has the following set of relations:

$$
\begin{aligned}
R = \{ & p^{10}a = ap, pacqr = rpcaq, ra = ar, p^{10}b = bp, p^2adq^2r = rp^2daq^2, \\
& rb = br, p^{10}c = cp, p^3bcq^3r = rp^3cbq^3, \\
& rc = cr, p^{10}d = dp, p^4bdq^4r = rp^4dbq^4, \\
& rd = dr, p^{10}e = ep, p^5ceq^5r = rp^5ecaq^5, re = er, \\
& aq^{10} = qa, p^6deq^6r = rp^6edbq^6pt = tp, \\
& bq^{10} = qb, p^7cdcq^7r = rp^7cdceq^7, \\
& qt = tq, cq^{10} = qc, p^8ca^3q^8r = rp^8a^3q^8, \\
& dq^{10} = qd, p^9da^3q^9r = rp^9a^3q^9, \\
& eq^{10} = qe, a^{-3}ta^3k = ka^{-3}ta^3\}.
\end{aligned}
$$

Let $G = \langle X \rangle$ be a finitely generated group. An automatic structure of $G$ with respect to $X$ is a set of finite-state automata:

- the word-acceptor, which accepts for every element of $G$ at least one word in $X^*$ representing it;
- multipliers, one for each $a \in X \cup \{1\}$, which accept a pair $(w_1, w_2)$, for words $w_i$ accepted by the word-acceptor, precisely when $w_1 a =_G w_2$.

# Automatic groups II

A group $G$ is said to be *biautomatic* if it has two multiplier automata, for left and right multiplication by elements of the generating set, respectively.

# Automatic groups II

A group $G$ is said to be *biautomatic* if it has two multiplier automata, for left and right multiplication by elements of the generating set, respectively.
Examples:

- Finite groups (consider the regular language to be the set of all words in the finite group).
- Example of biautomatic groups: hyperbolic groups, braid groups.

Automatic groups have solvable word problem in quadratic time.

Let $G = \langle X \mid R \rangle$, and suppose that $X$ is a symmetric set, that is $X = \{x, x^{-1} \mid x \in X\}$.

- The language associated to a finitely presented group is

$$L_G = \{w \in X^* \mid w =_G 1\}.$$

Let $G$ be a group.

**Theorem (Anisimov)**

*$L_G$ is a regular language if and only if $G$ is finite.*

**Theorem (Muller-Schupp)**

*$L_G$ is a context-free language if and only if $G$ is virtually free (i.e. $G$ has a free subgroup of finite index).*

# More about Anisimov's Theorem

- Let's try to prove one implication of the theorem, that is:

  If $G$ is finite, then $L_G$ is a regular language.
- In other words, let us construct a finite state automaton FSA that accepts $L_G$.
- The candidate is the Cayley graph of $G$.

Let $G$ be a group generated by a finite set $S = S^{-1}$. The Cayley graph $\Gamma(G, S)$ of $G$ is constructed as follows:

# Cayley graphs

Let $G$ be a group generated by a finite set $S = S^{-1}$. The Cayley graph $\Gamma(G, S)$ of $G$ is constructed as follows:

- Vertices: elements of $G$.

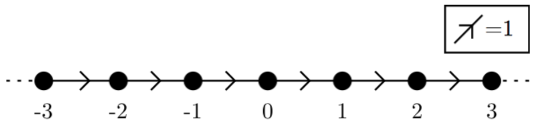
# Cayley graphs

Let $G$ be a group generated by a finite set $S = S^{-1}$. The Cayley graph $\Gamma(G, S)$ of $G$ is constructed as follows:

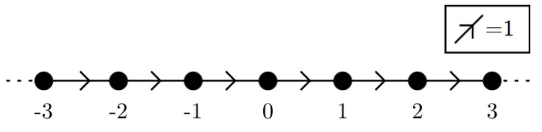
- Vertices: elements of $G$.
- Edges: for two vertices $x, y$, create the oriented edge $(x, y)$ if and only if there is some $s \in S$ such that $x \cdot s =_G y$. Then the edge $(x, y)$ is given the label $s$ and $(y, x)$ the label $s^{-1}$.

# Examples

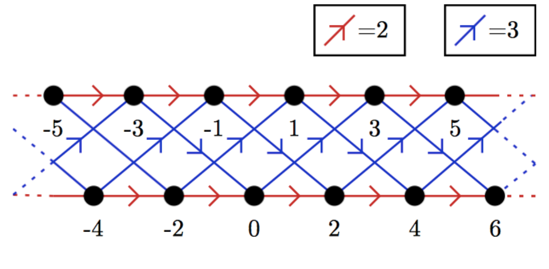
The Cayley graph of $\mathbb{Z}$, with respect to the generating set $S = \{1\}$.



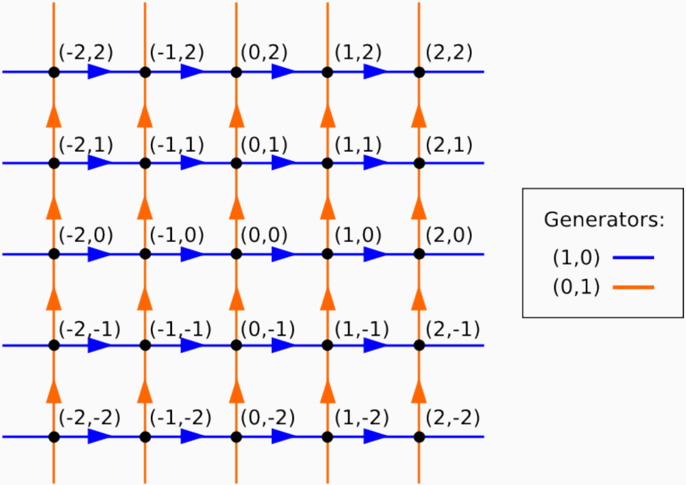
The Cayley graph of $\mathbb{Z}$, with respect to the generating set $S = \{1\}$.




The Cayley graph of $\mathbb{Z}$, with with respect to the generating set $S' = \{2, 3\}$.



The Cayley graph of $\mathbb{Z}^2$.



# About the implication of the Theorem

Let $G = \langle X \mid R \rangle$ be a group presentation.

- A word $w \in F(X)$ satisfies $w =_G 1$ (i.e. $w \in L_G$) if there exists a path in the Cayley graph of $G$ (with respect to $X$) with label $w$, starting and ending at the identity element (vertex) 1.

# About the implication of the Theorem

Let $G = \langle X \mid R \rangle$ be a group presentation.

- A word $w \in F(X)$ satisfies $w =_G 1$ (i.e. $w \in L_G$) if there exists a path in the Cayley graph of $G$ (with respect to $X$) with label $w$, starting and ending at the identity element (vertex) 1.
- If $G$ is finite, then the Cayley graph of $G$ is finite.

# About the implication of the Theorem

Let $G = \langle X \mid R \rangle$ be a group presentation.

- A word $w \in F(X)$ satisfies $w =_G 1$ (i.e. $w \in L_G$) if there exists a path in the Cayley graph of $G$ (with respect to $X$) with label $w$, starting and ending at the identity element (vertex) 1.
- If $G$ is finite, then the Cayley graph of $G$ is finite.
- The FSA is constructed out of a Cayley graph of $G$, by identifying the vertex representing the identity element as the start and accepting state.

## About the implication of the Theorem

Let $G = \langle X \mid R \rangle$ be a group presentation.

- A word $w \in F(X)$ satisfies $w =_G 1$ (i.e. $w \in L_G$) if there exists a path in the Cayley graph of $G$ (with respect to $X$) with label $w$, starting and ending at the identity element (vertex) 1.
- If $G$ is finite, then the Cayley graph of $G$ is finite.
- The FSA is constructed out of a Cayley graph of $G$, by identifying the vertex representing the identity element as the start and accepting state.
- Thus given any finite group, an FSA can be constructed in in this way from its Cayley graph.

Let $G = \langle X \mid R \rangle$ be a group presentation.

- A word $w \in F(X)$ satisfies $w =_G 1$ (i.e. $w \in L_G$) if there exists a path in the Cayley graph of $G$ (with respect to $X$) with label $w$, starting and ending at the identity element (vertex) 1.
- If $G$ is finite, then the Cayley graph of $G$ is finite.
- The FSA is constructed out of a Cayley graph of $G$, by identifying the vertex representing the identity element as the start and accepting state.
- Thus given any finite group, an FSA can be constructed in in this way from its Cayley graph.

Can you construct an FSA accepting the word problem in the Klein 4-group?


Take a break

In 1908, Dehn defined three famous decision problems for a finitely generated group $G = \langle X \mid R \rangle$.

- The word problem
- The conjugacy problem: asks whether or not two given input words $a, b \in F(X)$ are conjugate in $G$ (that is, there exists $g \in F(X)$ such that $g^{-1}ag =_G b$).

In 1908, Dehn defined three famous decision problems for a finitely generated group $G = \langle X \mid R \rangle$.

- The word problem
- The conjugacy problem: asks whether or not two given input words $a, b \in F(X)$ are conjugate in $G$ (that is, there exists $g \in F(X)$ such that $g^{-1}ag =_G b$).
- The isomorphism problem: for a class of groups $\mathcal{C}$ asks whether or not two given presentations of groups from $\mathcal{C}$ represent isomorphic groups. (decidable, for exammple, in finitely generated nilpotent groups [Grunewald-Segal])

# Groups in which the conjugacy problem is decidable

- free groups
- one-relator groups with torsion
- braid groups
- knot groups
- finitely generated abelian groups
- Gromov-hyperbolic groups

# Word problem (again!)

- In general, it may be difficult to work with a group $G$ defined by a presentation.
- It is complicated to discover whether or not a word represents the identity in the group.
- Formal languages and automata are used to study properties (i.e. the word problem, the isoperimetric functions) in many classes of important groups (for example, hyperbolic groups).

# If you are curious: Miscellanea

- Stallings automata: a simple and efficient algorithm for building a Deterministic Finite State Automaton associated to a given finitely generated subgroup of a free group.
  This DFSA is very useful: e.g. membership problem, intersection of subgroups, Nielsen-Schreier Theorem, etc.

- Stallings automata: a simple and efficient algorithm for building a Deterministic Finite State Automaton associated to a given finitely generated subgroup of a free group.
  This DFSA is very useful: e.g. membership problem, intersection of subgroups, Nielsen-Schreier Theorem, etc.
- Hyperbolic groups:

**Theorem**

Let $G$ be a finitely presented group. TFAE:

- $G$ is hyperbolic
- $G$ admits a linear isoperimetric function
- $G$ admits a subquadratic isoperimetric function

# Automata groups

# Motivation

The class of automata groups contains several remarkable countable groups. They have applications in several areas of mathematics (algebra, geometry, analysis, probability, etc.)

With respect to group theory, they have been used to solve many big problems.

# Mealy machine

Informally:

- A *finite state transducer* (FST) is a finite state automaton with an output function.
- An FST is *deterministic* if the corresponding FSA is deterministic (ignoring the output function).
- An FST is also known as a Mealy machine.

Formally, an FST is defined with:

- A finite set of states $Q$.
- A finite set of input symbols $A$.
- A finite set of output symbols $O$.
- An input transition function $\delta : Q \times A \to Q$.
- An output transition function $X : Q \to O$.
- An initial state $q_0 \in Q$ from which every process starts.

We will use the following notation:

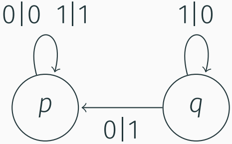
$$\mathcal{M} = (Q, A, O, \delta, X, q_0).$$

- Each transition (edge) is labelled with two strings as follows

$$input|output$$

# How to generate automata groups

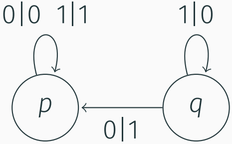
Consider the alphabet $A = \{0, 1\}$, and two states $p$ and $q$.




- Take the string $w = 010$.
- How can we read $w$ starting at $p$?
- And at $q$?

# Let's start with an example

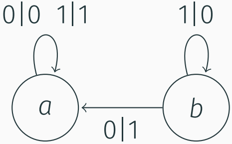
Consider the alphabet $A = \{0, 1\}$, and two states $p$ and $q$.




- Take the string $w = 010$.
- How can we read $w$ starting at $p$?
- And at $q$?

Since we can do this for any word, we can say that $p$ induces a map $\rho_p : A^* \to A^*$. Similarily, $q$ induces a map $\rho_q : A^* \to A^*$.

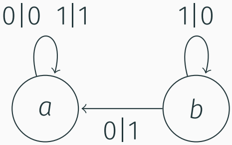
# Example



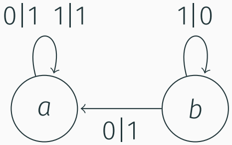
# How to generate automata groups

- We get a map $\rho_q$ from $A^*$ to $A^*$ per state $q \in Q$.
- These maps have the same domain and codomain, hence we can compose them as we want.
- Note that the composition corresponds in the automaton to plugging in the output of a run to the input of another run.
- Given two states $p, q \in Q$, we define $\rho_{pq} = \rho_p \circ \rho_q$.
- The structure generated by $\{\rho_q \mid q \in Q\}$ is a semigroup.
- If the map $\rho_q$ is bijective for all $q \in Q$, i.e. every state induces a function that can be inverted, then we obtain an *automata group*.

# Let's go back to the example




What is the group generated by the automaton above?




What is the group generated by the automaton above?

# The Gupta-Sidki automaton

# The Gupta-Sidki automaton

Sorry, you have to wait until Thursday ;)

The Grigorchuk and the Gupta-Sidki automata are examples of *self-similar* group (we will see the definition later).

Self-similar groups are a subclass of the class of groups of automorphisms of rooted trees.

# Let's start now with Part III: groups of automorphisms of rooted trees

# Motivation: famous problems in group theory

- Milnor's Problem $\Longrightarrow$ growth of a group.
- General Burnside Problem $\Longrightarrow$ finiteness properties of a group.

Let $G = \langle X \mid R \rangle$ be a presentation of a group $G$.

For each $g \in G$, let $|g|$ denote the smallest length of a word $w \in F(X)$ such that $w =_G g$.

The *growth function* of $G$ is the map $\gamma : \mathbb{N} \to \mathbb{N}$:

$$\gamma(n) = |\{g \in G \mid |g| = n\}|.$$

It depends on the chosen presentation $\langle X \mid R \rangle$.

# Some facts about growth of groups

- Free groups of finite rank $k > 1$: exponential growth.

## Some facts about growth of groups

- Free groups of finite rank $k > 1$: exponential growth.
- Fundamental group $\pi_1(M)$ of a closed negatively curved Riemannian manifold: exponential growth.

## Some facts about growth of groups

- Free groups of finite rank $k > 1$: exponential growth.
- Fundamental group $\pi_1(M)$ of a closed negatively curved Riemannian manifold: exponential growth.
- Finite groups: polynomial growth (of degree 0).

# Some facts about growth of groups

- Free groups of finite rank $k > 1$: exponential growth.
- Fundamental group $\pi_1(M)$ of a closed negatively curved Riemannian manifold: exponential growth.
- Finite groups: polynomial growth (of degree 0).
- (Gromov, 1981) A group is virtually nilpotent if and only if it has polynomial growth.

# Some facts about growth of groups

- Free groups of finite rank $k > 1$: exponential growth.
- Fundamental group $\pi_1(M)$ of a closed negatively curved Riemannian manifold: exponential growth.
- Finite groups: polynomial growth (of degree 0).
- (Gromov, 1981) A group is virtually nilpotent if and only if it has polynomial growth.

Milnor's question (1960):

*Are there groups of intermediate growth between polynomial and exponential?*

- Free groups of finite rank $k > 1$: exponential growth.
- Fundamental group $\pi_1(M)$ of a closed negatively curved Riemannian manifold: exponential growth.
- Finite groups: polynomial growth (of degree 0).
- (Gromov, 1981) A group is virtually nilpotent if and only if it has polynomial growth.

Milnor's question (1960):

*Are there groups of intermediate growth between polynomial and exponential?*

Grigorchuk's answer (1980):

*Yes, the first . . . Grigorchuk group.*

# About the General Burnside Problem

*A still undecided point in the theory of discontinuous groups is whether the order of a group may be not finite, while the order of every operation it contains is finite.*

W. Burnside (1902)

*A still undecided point in the theory of discontinuous groups is whether the order of a group may be not finite, while the order of every operation it contains is finite.*

W. Burnside (1902)

In modern terminology the general Burnside problem asks:

can a finitely generated periodic group be finite?

Recall that a group $G$ is periodic if for any $g \in G$ there exists a positive integer $n$ such that $g^n = 1$.

# Are finitely generated periodic groups finite?

- Yes, for nilpotent groups.
- Yes, for finitely generated periodic subgroups of the general linear group of degree $n > 1$ over the complex numbers.
- Yes, … for many other classes of groups.

# Are finitely generated periodic groups finite?

- Yes, for nilpotent groups.
- Yes, for finitely generated periodic subgroups of the general linear group of degree $n > 1$ over the complex numbers.
- Yes, . . . for many other classes of groups.
- Counterexample: the first Grigorchuk group, the Gupta-Sidki $p$-groups.

# Plan for Thursday

# Plan for Thursday

- Automorphisms of regular rooted trees
- Self-similar groups
- Self-similar groups as Mealy automata
- Branch groups
- Examples
    - The Grigorchuk groups
    - The Gupta-Sidki group
    - The GGS-groups
    - The Basilica group
    - The Hanoi Tower group

Questions or answers?

Obrigada :)