

# Formal languages and group theory

Rick Thomas



**University of  
Leicester**

Department of Computer Science

<http://www.cs.le.ac.uk/people/rthomas/>

# Lecture 1

## Formal language theory

# Languages and monoids

$\Sigma$  : finite set of symbols.

$\Sigma^+$  : the set of all non-empty finite words formed from the symbols in  $\Sigma$ .

**Example.**  $\Sigma = \{ a, b \};$

$\Sigma^+ = \{ a, b, a^2, ab, ba, b^2, a^3, a^2b, \dots \}.$

$\Sigma^+$  forms a semigroup (under the operation of concatenation):

- *associative*  $(\alpha\beta)\gamma = \alpha(\beta\gamma)$  for all  $\alpha, \beta, \gamma$ .

$\Sigma^*$  : the set of all finite words formed from the symbols in  $\Sigma$   
(including the *empty word*  $\varepsilon$ ).

**Example.**  $\Sigma = \{ a, b \}$ ;  $\Sigma^* = \{ \varepsilon, a, b, a^2, ab, ba, b^2, a^3, a^2b, \dots \}$ .

$\Sigma^*$  forms a monoid (under the operation of concatenation):

- *associative*  $(\alpha\beta)\gamma = \alpha(\beta\gamma)$  for all  $\alpha, \beta, \gamma$ .
- *identity*  $\varepsilon$   $\alpha\varepsilon = \varepsilon\alpha = \alpha$  for all  $\alpha$ .

$\Sigma^+$  is the *free semigroup* and  $\Sigma^*$  is the *free monoid* on the set  $\Sigma$ .

We often refer to the operation in a semigroup or monoid as a *product*.

Let  $S$  be a semigroup. We say that a subset  $\Sigma$  of  $S$  *generates*  $S$  if every element of  $S$  is a product of elements of  $\Sigma$ .

If  $S$  is a semigroup generated by a finite set  $\Sigma$ , then every element of  $S$  is expressible as an element of  $\Sigma^+$ .

If  $S$  is a semigroup generated by a finite set  $\Sigma$  then there is a *homomorphism*  $\varphi : \Sigma^+ \rightarrow S$  (i.e. we have  $(\alpha\varphi)(\beta\varphi) = (\alpha\beta)\varphi$  for all words  $\alpha$  and  $\beta$  in  $\Sigma^+$ ).

In this case, the semigroup  $S$  is isomorphic to  $\Sigma^+ / \approx$  where  $\approx$  is the *congruence* (equivalence relation preserved under concatenation) on  $\Sigma^+$  defined by

$$\alpha \approx \beta \iff \alpha\varphi = \beta\varphi.$$

If  $\mathcal{R}$  is a set of *relations* on a semigroup  $S$  generated by  $\Sigma$  (i.e. if  $\mathcal{R}$  is a set of equations of the form  $\alpha = \beta$  where  $\alpha, \beta \in \Sigma^+$  and where  $\alpha$  and  $\beta$  represent the same element of  $S$ ), then  $\mathcal{R}$  generates a congruence  $\approx$  on  $\Sigma^+$  (where  $\approx$  is the smallest congruence on  $\Sigma^+$  containing  $\mathcal{R}$ ).

We say that  $\mathcal{R}$  is a set of *defining relations* for  $S$  if  $S$  is isomorphic to  $\Sigma^+ / \approx$ .

This is effectively saying that every relation which holds in  $S$  is a consequence of the relations in  $\mathcal{R}$ .

We then say that  $\langle \Sigma : \mathcal{R} \rangle$  is a *presentation* for the semigroup  $S$ .

**Example.**

If we have the presentation

$$\langle a, b : a^2 = a, b^2 = b \rangle$$

for a semigroup  $S$ , then every element of  $S$  is equal to a word of the form:

$$\begin{array}{ll} abab....ab, & abab....ba, \\ baba....ba, & \text{or} \quad baba....ab. \end{array}$$

The free semigroup  $\Sigma^+$  on a set  $\Sigma$  has the presentation  $\langle \Sigma : \rangle$ .

We have a similar idea for monoids. A subset  $\Sigma$  of a monoid  $M$  *generates*  $M$  if every non-identity element of  $M$  is a product of elements of  $\Sigma$ . If  $M$  is generated by a finite set  $\Sigma$ , then every element of  $M$  is expressible as an element of  $\Sigma^*$ . The empty word  $\varepsilon$  represents the identity of  $M$ .

There is a homomorphism  $\varphi : \Sigma^* \rightarrow M$  and  $M \cong \Sigma^* / \approx$  where  $\approx$  is the congruence on  $\Sigma^*$  defined by  $\alpha \approx \beta \Leftrightarrow \alpha\varphi = \beta\varphi$ .

If  $\mathcal{R}$  is a set of relations on a monoid  $M$ , then  $\mathcal{R}$  generates a congruence  $\approx$  on  $\Sigma^*$ . We say that  $\mathcal{R}$  is a set of *defining relations* for  $M$  if  $M$  is isomorphic to  $\Sigma^* / \approx$ .

We then say that  $\langle \Sigma : \mathcal{R} \rangle$  is a *presentation* for  $M$ .



**Example.** If we have the presentation

$$\langle a, b : a^2 = a, b^2 = b \rangle$$

for a monoid  $M$ , then every non-identity element of  $M$  is equal to a word of the form:

$$abab....ab, \quad abab....ba,$$

$$baba....ba, \quad \text{or} \quad baba....ab.$$

The identity element is represented by  $\varepsilon$  (and by no other word).

The free monoid  $\Sigma^*$  on a set  $\Sigma$  has the presentation  $\langle \Sigma : \quad \rangle$ ; note that this is a monoid (as opposed to a semigroup) presentation.

# Languages

A *language*  $L$  is a subset of  $\Sigma^*$  (for some finite set  $\Sigma$ ).

**Example.**  $\Sigma = \{ a, b \}$   $L = \{ \alpha \in \Sigma^* : |\alpha| \text{ is even} \}$ .

**Example.**  $\Sigma = \{ a, b, c \}$   $L = \{ a^n b c^n : n \in \mathbf{N} \}$ .

Note that the set  $\mathbf{N}$  of natural numbers is taken to contain 0 here.

Some classes of languages and the associated “machines”:

<i>Languages</i>	<i>Machines</i>
Regular languages	Finite automata
Context-free languages	Pushdown automata
Recursive languages	Turing machines
Recursively enumerable languages	

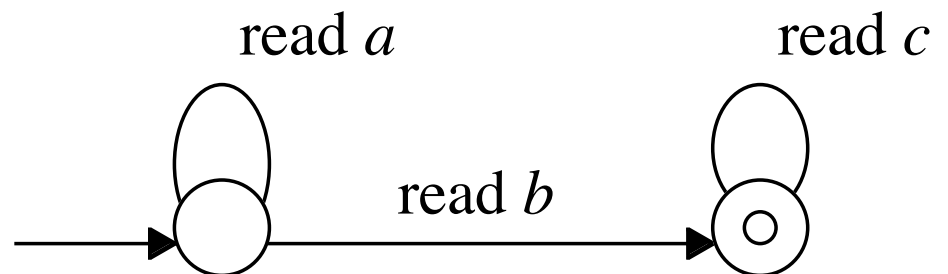
# Regular languages.

Regular languages are the languages accepted by *finite automata*.

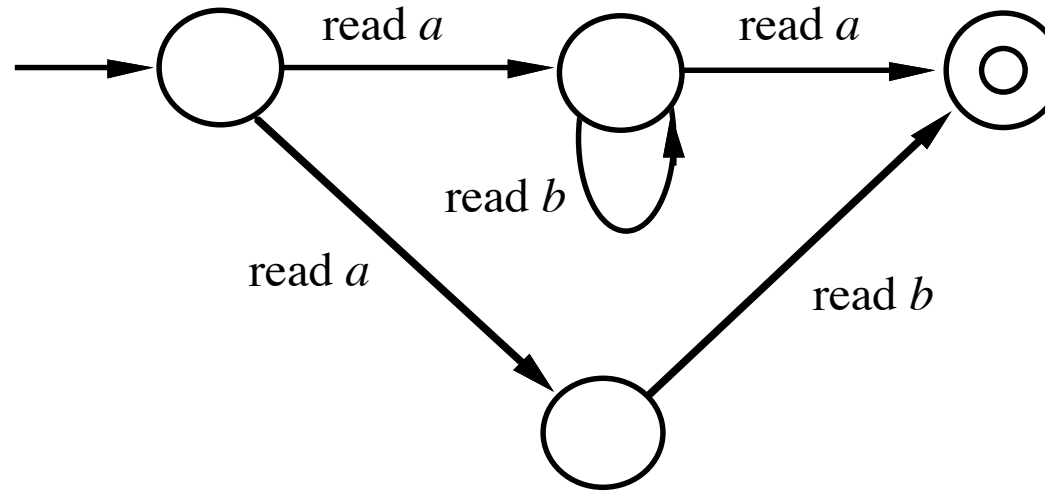
Finite automata have states with a designated *start state* and a set of *accept states*. A word  $\alpha$  is *accepted* by an automaton  $M$  if  $\alpha$  maps the start state to an accept state.

For example, the finite automaton below accepts the language

$$\{a^n b c^m : n, m \in \mathbf{N}\}:$$



Allowing non-determinism, such as



does not increase the range of languages accepted.

In a non-deterministic machine, a word is accepted if at least one computation path leads to acceptance; in our example, the machine accepts the language  $\{ab^n a : n \in \mathbf{N}\} \cup \{ab\}$ .

## Some other definitions of regular languages.

*Regular grammars*  $(N, \Sigma, P, S)$

We have a set  $N$  of *non-terminals* that can be rewritten.

We have a set  $\Sigma$  of *terminals* that cannot be rewritten.

We have a set  $P$  of *production rules*.

Each production rule is of the form  $A \rightarrow xB$  where  $A, B \in N$  and  $x \in \Sigma$  or else of the form  $A \rightarrow \varepsilon$ .

There is a designated starting symbol  $S \in N$ . The *language* of the grammar is the set of all words in  $\Sigma^*$  that can be derived from  $S$ .

**Example.** Consider the regular grammar

$$S \rightarrow aS \mid bT \quad T \rightarrow cT \mid \varepsilon$$

where  $N = \{S, T\}$  and  $\Sigma = \{a, b, c\}$ .

The starting symbol is  $S$ .

Starting from  $S$  we can proceed as follows:

$$\begin{aligned} S &\rightarrow aS \rightarrow a^2S \rightarrow \dots \rightarrow a^nS \rightarrow a^nbT \rightarrow a^nbcT \rightarrow a^nbc^2T \rightarrow \\ &\dots \rightarrow a^nbc^mT \rightarrow a^nbc^m. \end{aligned}$$

This grammar generates the regular language

$$\{a^nbcm : n, m \in \mathbf{N}\}.$$

## *Rational expressions*

$\emptyset$ ,  $\{\varepsilon\}$  and  $\{a\}$  (for any  $a \in \Sigma$ ) are regular languages.

If  $K$  and  $L$  are regular language then so is  $K \cup L$ .

If  $K$  and  $L$  are regular languages then so is  $KL = \{ab : a \in K, b \in L\}$ .

If  $K$  is a regular languages then so is  $K^* = \{a_1a_2\dots a_n : a_i \in K, n \in \mathbf{N}\}$ .

So any language which can be built up from  $\emptyset$ ,  $\{\varepsilon\}$  and  $\{a\}$  by means of union, concatenation and star is regular.

For example, the rational expression  $a^*bc^*$  represents the language

$$\{a^nbc^m : n, m \in \mathbf{N}\}.$$



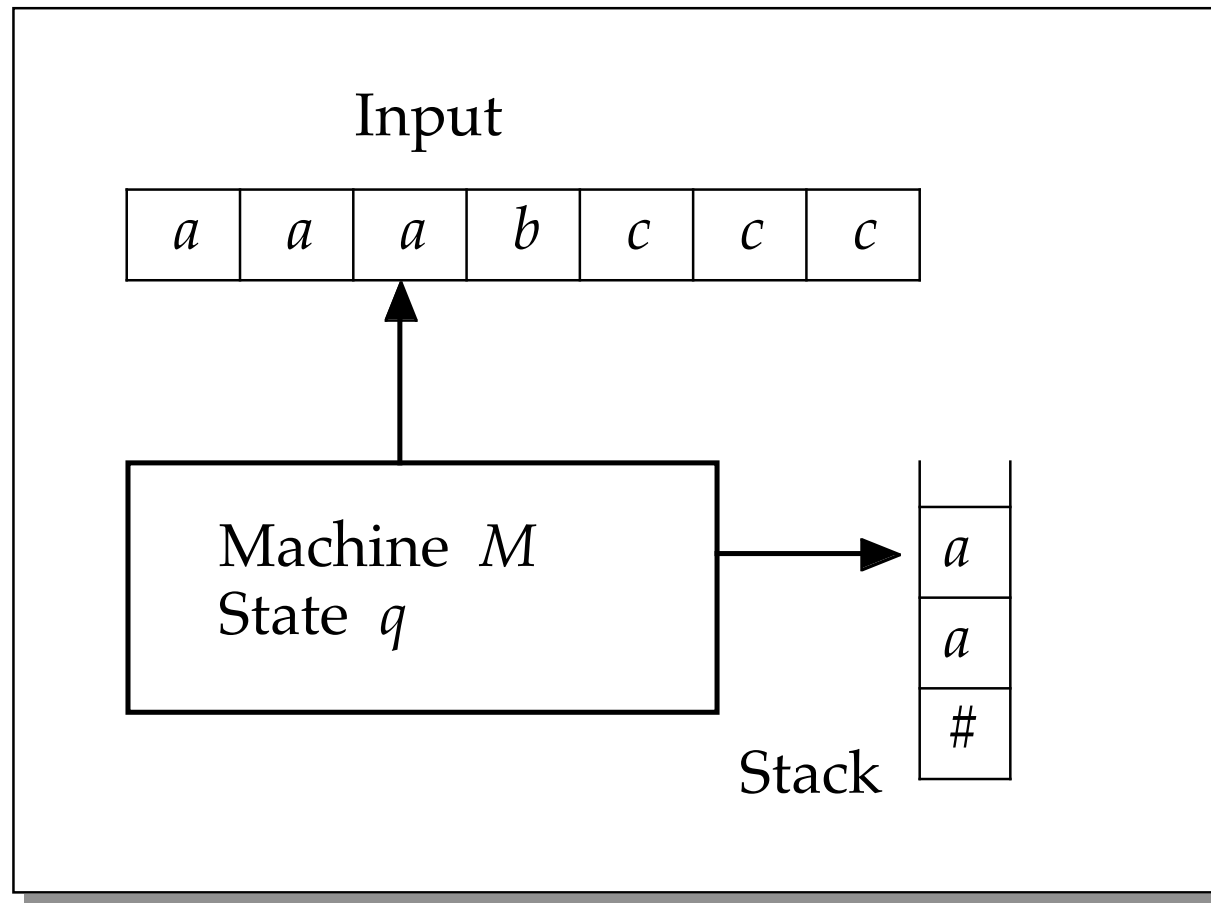
# Context-free languages

We may extend a (non-deterministic) finite automaton by adding a *stack* to get a *pushdown automaton*.

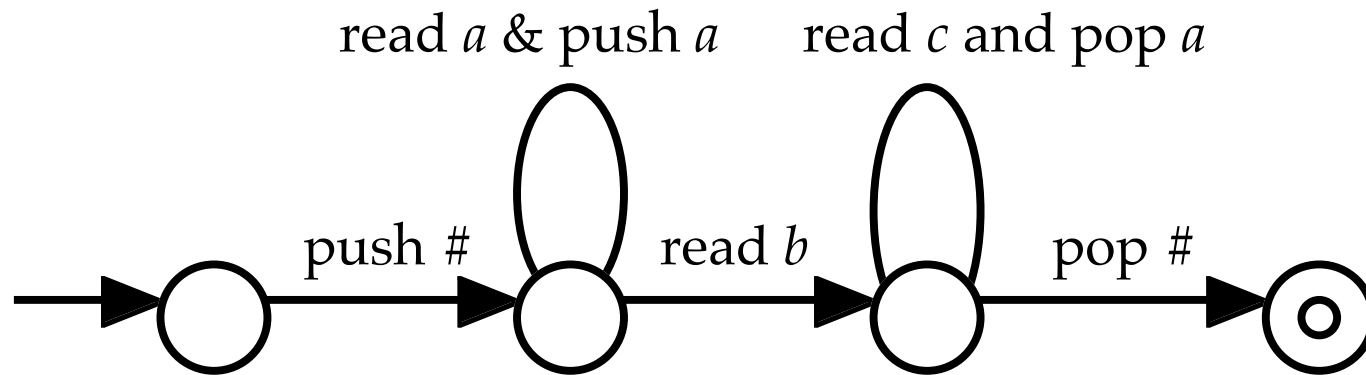
Again, there is a designated *start state* and a set of *accept states*.

A word is *accepted* if one can reach an accept state when all the input has been read. We start with an empty stack and we don't worry about what is on the stack at the end of the computation.

The languages accepted by pushdown automata are known as *context-free languages*.



For example, the pushdown automaton shown below accepts the language  $\{a^nbc^n : n \in \mathbf{N}\}$ .



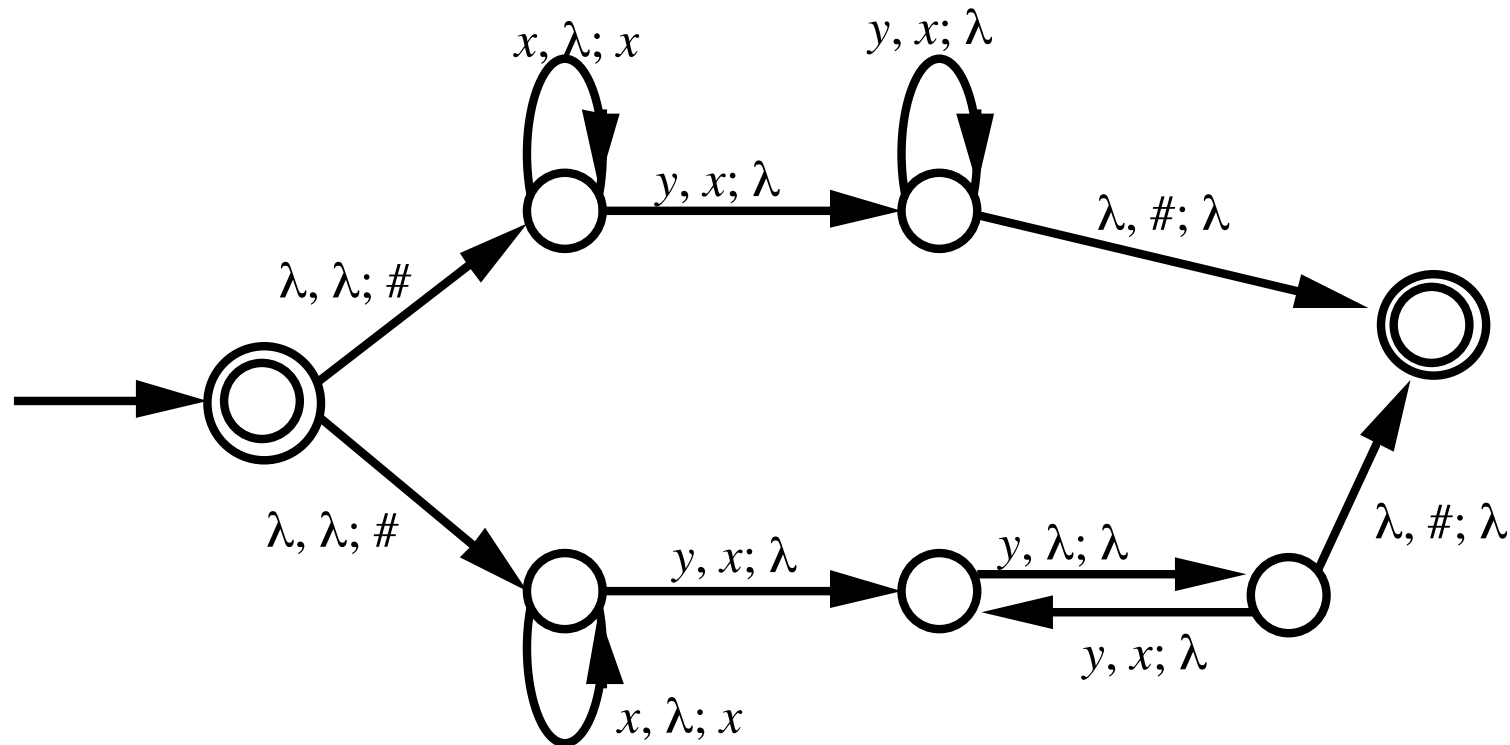
We often label the edges by  $x, y; z$  (read  $x$ , pop  $y$ , push  $z$ ).

Insisting that the machine is deterministic does restrict the range of languages accepted in this case (the class of *deterministic context-free languages*).

The machine shown in the above example is deterministic.

The pushdown automaton shown below accepts the language

$$L = \{x^n y^n : n \in \mathbf{N}\} \cup \{x^n y^{2n} : n \in \mathbf{N}\}.$$



$L$  is not accepted by any deterministic pushdown automaton.

## Another definition of context-free languages.

In a similar fashion to regular grammars we have *context-free grammars*; the only difference to regular grammars that there is now no restriction on the right-hand side of a production rule (i.e. each production rule is of the form  $A \rightarrow \alpha$  for some  $\alpha \in (N \cup \Sigma)^*$ ).

**Example.**  $S \rightarrow aSc \mid b$

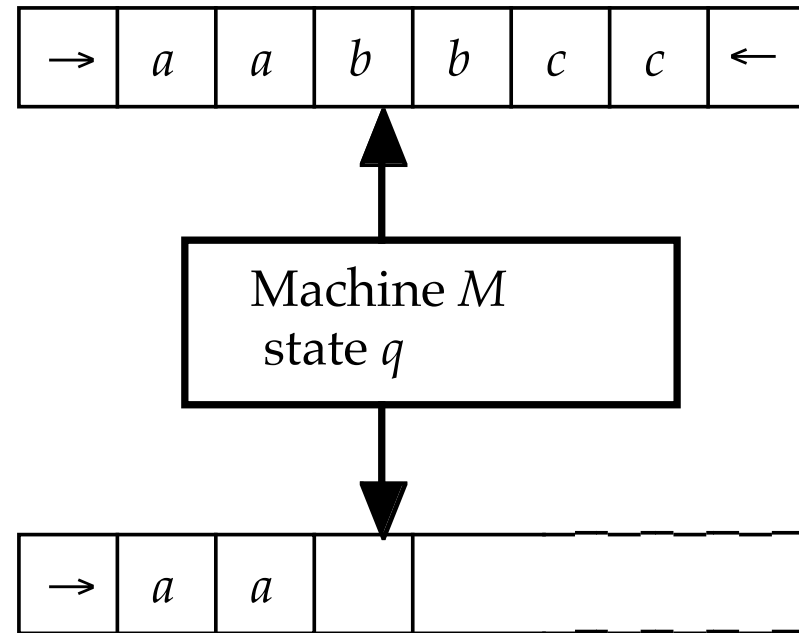
This context-free grammar generates the language  $\{a^nbc^n : n \in \mathbf{N}\}$ :

$$S \rightarrow aSb \rightarrow a^2Sc^2 \rightarrow \dots \rightarrow a^nSc^n \rightarrow a^nbc^n.$$

# Turing machines

A most general model of computation is the *Turing machine*.

Here we have some memory (in the form of a *work tape*) as well as the input.



We also have (at least one) *halt state* (as opposed to accept states).

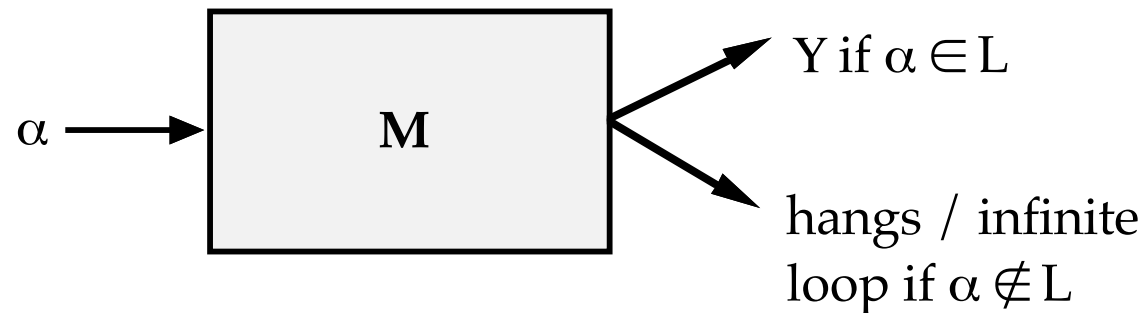
A Turing machine can move freely over its input tape. It can write symbols to its work tape (and erase them).

A Turing machine with a given input  $\alpha$  will either

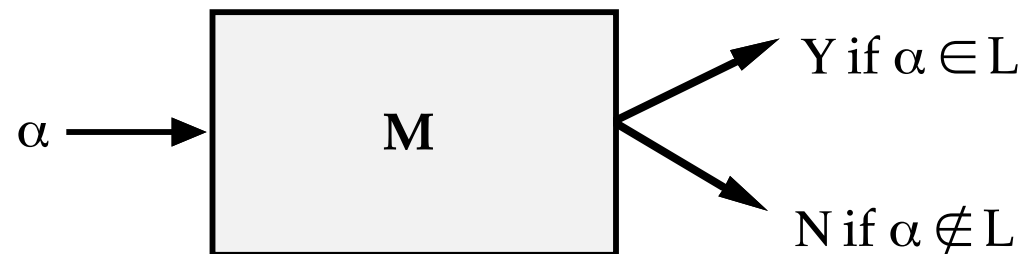
- (i) terminate (if it enters a halt state); or
- (ii) hang (no legal move defined); or
- (iii) run indefinitely without terminating or hanging (sometimes referred to as an “infinite loop”).

There are two main classes of languages associated with Turing machines.

*Recursively enumerable* (simple acceptance – one halt state)



*Recursive* (decision process – “accept” and “reject” halt states)





Any recursive language is recursively enumerable but the converse is false.

What about non-deterministic Turing machines?

*Recursively enumerable:* the Turing machine accepts the input if it is possible to enter the halt state.

*Recursive:* the Turing machine accepts the input if some computation path leads to the accepting halt state and rejects the input if every computation path leads to the rejecting halt state.

Allowing non-determinism does not increase the range of languages accepted or decided by Turing machines.

However, if we put restrictions on the amount of time or space allowed for the computation, the situation may change.

**P = NP Question.**

Is the class **P** of languages decided by deterministic Turing machines in polynomial time equal to the class **NP** of languages decided by non-deterministic Turing machines in polynomial time ?

# Lecture 2

## Group theory

# Groups

*Group:* A monoid  $G$  such that every element  $g$  has an *inverse*  $g^{-1}$  :  
 $gg^{-1} = g^{-1}g = 1$  for all  $g \in G$  (where 1 is the identity element).

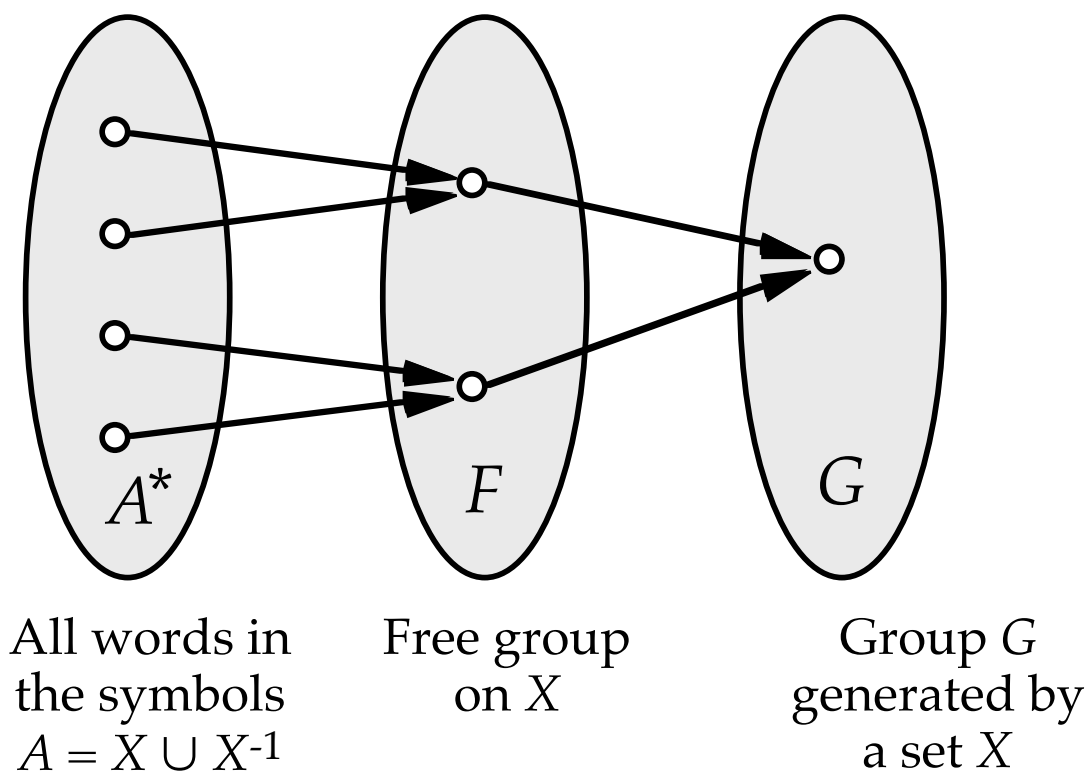
Let  $G$  be a group generated by a finite set  $X$  so that each element of  $G$  may be written as a word in the symbols  $x, x^{-1}$  (where  $x \in X$ ).  
Let  $A = X \cup X^{-1}$ .

Every element of  $G$  may be expressed as a *reduced word* (one which contains no subwords of the form  $xx^{-1}$  or  $x^{-1}x$ ) over  $A$ .

The set of all reduced words on  $A$  forms the *free group* on  $X$ .

If  $G$  is a group generated by  $X$  and  $F$  is the free group on  $X$  then the map from  $A^*$  to  $G$  factors through the map onto  $F$ :

$$A^* \rightarrow F \rightarrow G.$$



If  $G$  is generated by a finite set  $X$  and  $A = X \cup X^{-1}$  then  $G$  is isomorphic to  $A^* / \approx$ , where  $\approx$  is some congruence on  $\Sigma^*$  containing every pair of the form  $(xx^{-1}, \varepsilon)$  or  $(x^{-1}x, \varepsilon)$  with  $x \in X$ .

In general each element of the group is represented by several (reduced) words in  $A^*$ .

A word  $\alpha$  is said to be *cyclically reduced* if it is not of the form  $x^{-1}\beta x$  or  $x\beta x^{-1}$  for any  $x \in X$  and any word  $\beta$ .

# Group presentations

The idea here is similar to semigroups and monoids but we must take care of inverses. Note that words  $\alpha$  and  $\beta$  represent the same element of a group  $G$  if and only if  $\alpha\beta^{-1}$  represents the identity element of  $G$ .

A *presentation* for a group  $G$  is an expression of the form  $\langle X : \mathcal{R} \rangle$  where  $\mathcal{R}$  is a set of defining *relators*, i.e. a set of words such that  $G$  is  $A^*/\approx$ , where  $\approx$  is the congruence on  $A^*$  generated by pairs of the form  $(xx^{-1}, \varepsilon)$  or  $(x^{-1}x, \varepsilon)$  with  $x \in X$  and pairs of the form  $(\alpha, \varepsilon)$  with  $\alpha \in \mathcal{R}$ . We may take the words in  $\mathcal{R}$  to be cyclically reduced.

*Example.*  $G = \langle a, b : aba^{-1}b^{-1} \rangle$ .

The words  $a, bab^{-1}, b^{-1}ab, bbab^{-1}a^{-1}ab^{-1}, \dots$  all represent the same element of  $G$ .

A group  $G$  is said to be *finitely generated* if it has a finite generating set  $X$  (i.e. every element of  $G$  is a product of elements of  $X$  and the inverses of elements of  $X$ ; notice that this is a *group generating set* as opposed to a *monoid generating set*).

A group  $G$  is said to be *finitely presented* if it has a presentation  $\langle X : \mathcal{R} \rangle$  with both  $X$  and  $\mathcal{R}$  finite. If  $\mathcal{R}$  is empty then we have a presentation for the free group on  $X$ .



**Example.**  $G = \langle a, b : a^3, b^2, (ab)^2 \rangle$ .

$G$  is a group with six elements. The words

$$\varepsilon, a, a^2, b, ab, a^2b$$

represent the six elements of  $G$ .

**Example.**  $G = \langle a, b : aba^{-1}b^{-1} \rangle$ .

$G$  is an infinite group. Let  $A$  denote  $a^{-1}$  and  $B$  denote  $b^{-1}$ . Each element of  $G$  is represented by a word of the form  $a^i b^j$ ,  $A^i b^j$ ,  $a^i B^j$  or  $A^i B^j$  (with  $i, j \in \mathbf{N}$ ).

If  $\langle X : \mathcal{R} \rangle$  is a presentation for a group  $G$  and if  $\alpha$  is any reduced non-empty word representing the identity element of  $G$ , then we can express  $\alpha$  in the form

$$\beta_1^{-1}r_1\beta_1 \cdot \beta_2^{-1}r_2\beta_2 \cdot \dots \cdot \beta_s^{-1}r_s\beta_s$$

for some words  $\beta_1, \beta_2, \dots, \beta_s$  over  $X \cup X^{-1}$  and for some words  $r_1, r_2, \dots, r_s$  in  $\mathcal{R} \cup \mathcal{R}^{-1}$ .

Note that  $\alpha$  is equal to  $\beta_1^{-1}r_1\beta_1 \cdot \beta_2^{-1}r_2\beta_2 \cdot \dots \cdot \beta_s^{-1}r_s\beta_s$  in the free group on  $X$  (i.e. to get  $\alpha$  from  $\beta_1^{-1}r_1\beta_1 \cdot \beta_2^{-1}r_2\beta_2 \cdot \dots \cdot \beta_s^{-1}r_s\beta_s$  we simply need to continually delete subwords of the form  $xx^{-1}$  or  $x^{-1}x$ ).

If  $G$  is a group,  $H \subseteq G$  and  $H$  forms a group, then  $H$  is said to be a *subgroup* of  $G$ .

We write  $H \leq G$  if the group  $H$  is a subgroup of the group  $G$ . The group  $G$  is then said to be an *overgroup* of the group  $H$ .

If  $H$  is a subgroup of  $G$  and  $g \in G$ , then the set  $Hg = \{hg : h \in H\}$  is said to be a *coset* of  $H$  in  $G$ .

The group  $G$  is the disjoint union of the cosets of  $H$ . The number of cosets is called the *index* of  $H$  in  $G$ . If there are only finitely many such cosets, then we say that  $H$  has *finite index* in  $G$ .

A group  $G$  that does not have a non-trivial proper congruence on it (i.e. has no non-trivial proper quotients) is said to be *simple*.

A congruence  $\approx$  on a group  $G$  corresponds to a *normal* subgroup  $N$  (i.e. to a subgroup  $N$  of  $G$  such that  $g^{-1}Ng = N$  for all  $g \in G$ ). We then have that  $g \approx h$  if and only if  $Ng = Nh$ .

Factoring out the congruence  $\approx$  is equivalent to taking the factor group  $G/N$  with elements the cosets  $\{Ng : g \in G\}$  and operation defined by  $(Ng)(Nh) = N(gh)$ .

## Group constructions

If  $G$  and  $K$  are groups with presentations  $\langle X : \mathcal{R} \rangle$  and  $\langle Y : S \rangle$  with  $X \cap Y = \emptyset$ , we write  $G * K$  for the *free product* and  $G \times K$  for the *direct product* of  $G$  and  $K$ , i.e. the groups with presentations

$$\langle X \cup Y : \mathcal{R} \cup S \rangle \quad \text{and}$$

$$\langle X \cup Y : \mathcal{R} \cup S \cup \{x^{-1}y^{-1}xy : x \in X, y \in Y\} \rangle$$

respectively. A finitely generated free group is then a group of the form  $\mathbf{Z} * \mathbf{Z} * \dots * \mathbf{Z}$ .

# Subsets of monoids and groups

$M$  - finitely generated monoid.  $S \subseteq M$ .

$\varphi : \Sigma^* \rightarrow M$  - homomorphism.  $L = S\varphi^{-1}$ .

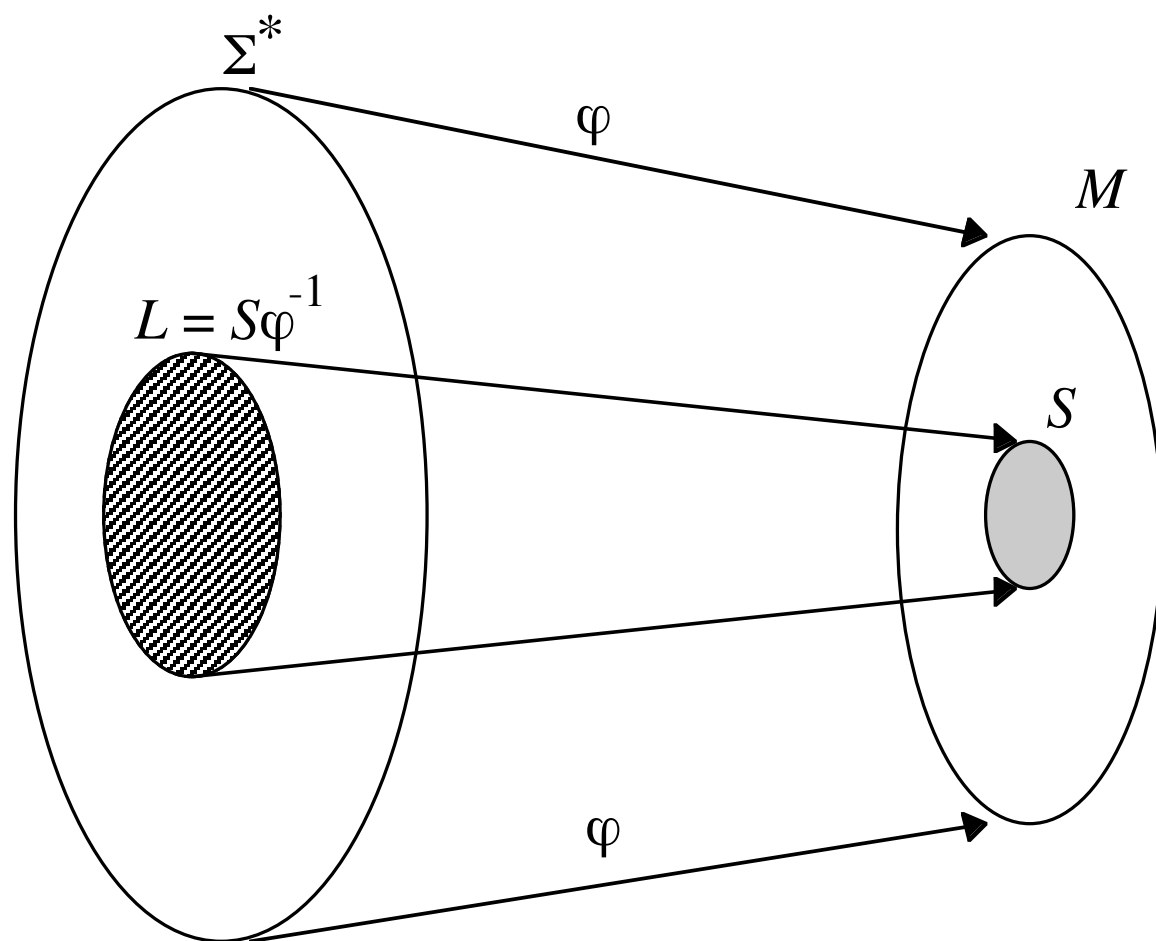
If  $\mathcal{F}$  is a family of languages then we say that  $\mathcal{F}$  is *closed under*:

- *homomorphisms* if, whenever  $L \in \mathcal{F}$ ,  $L \subseteq \Sigma^*$  and  $\varphi : \Sigma^* \rightarrow \Omega^*$  is a homomorphism, then  $L\varphi \in \mathcal{F}$ ;
- *inverse homomorphisms* if, whenever  $L \in \mathcal{F}$ ,  $L \subseteq \Omega^*$  and  $\varphi : \Sigma^* \rightarrow \Omega^*$  is a homomorphism, then  $L\varphi^{-1} \in \mathcal{F}$ .

If  $\mathcal{F}$  is a family of languages then we say that  $\mathcal{F}$  is *closed under*:

- *intersection with regular languages* if, whenever  $L \in \mathcal{F}$ ,  $L \subseteq \Sigma^*$ ,  $K$  is regular and  $K \subseteq \Sigma^*$ , then  $L \cap K \in \mathcal{F}$ .

If  $\mathcal{F}$  is a family of languages which is closed under inverse homomorphism,  $M$  is a monoid, and  $\varphi : \Sigma^* \rightarrow M$  and  $\psi : \Omega^* \rightarrow M$  are homomorphisms, then  $S\varphi^{-1} \in \mathcal{F}$  if and only if  $S\psi^{-1} \in \mathcal{F}$ .





A subset  $S$  of a monoid  $M$  is said to be a *recognizable* subset of  $M$  if  $L = S\varphi^{-1}$  is a regular subset of  $\Sigma^*$ . We write  $S \in \mathcal{R}ec(M)$ .

Let  $M$  be a finitely generated monoid and let  $S \subseteq M$ . Then the following are equivalent:

- (i)  $S = L\varphi$  for some regular  $L \subseteq \Sigma^*$ .
- (ii)  $S$  is generated by a regular grammar over  $M$ .
- (iii)  $S$  is represented by a rational expression over  $M$ .

We say that  $S$  is a *rational subset* of  $M$  in this case and we write  $S \in \mathcal{R}at(M)$ .

$S$  is said to be a *context-free* subset of  $M$  if  $L = S\varphi^{-1}$  is a context-free subset of  $\Sigma^*$ . We write  $S \in CF(M)$ .

Let  $M$  be a finitely generated monoid and let  $S \subseteq M$ . Then the following are equivalent:

- (i)  $S = L\varphi$  for some context-free  $L \subseteq \Sigma^*$ .
- (ii)  $S$  is generated by a context-free grammar over  $M$ .

We say that  $S$  is an *algebraic subset* of  $M$  in this case and we write  $S \in Alg(M)$ .

Clearly

$$\mathcal{R}ec(M) \subseteq \mathcal{R}at(M) \quad \text{and} \quad \mathcal{C}\mathcal{F}(M) \subseteq \mathcal{A}lg(M).$$

$$\mathcal{R}ec(M) \subseteq \mathcal{C}\mathcal{F}(M) \quad \text{and} \quad \mathcal{R}at(M) \subseteq \mathcal{A}lg(M).$$

If  $G$  is a group and  $S$  is a subset of  $G$  then  $S \in \mathcal{R}ec(G)$  if and only if  $S$  is a union of cosets of a subgroup of finite index.

**Anisimov & Seifert.** If  $G$  is a group and  $H$  is a subgroup of  $G$  then  $H \in \mathcal{R}at(G)$  if and only if  $H$  is finitely generated.

In a finitely generated monoid  $M$  we have that

$$\mathcal{R}ec(M) \subseteq \mathcal{R}at(M).$$

$M$  is said to be a *Kleene monoid* if and only if  $\mathcal{R}ec(M) = \mathcal{R}at(M)$ .

A group  $G$  is a Kleene monoid if and only if it is finite.

A monoid  $M$  in which  $\mathcal{C}\mathcal{F}(M) = \mathcal{A}lg(M)$  is said to be *algebraic*.

**Question.** Which groups are algebraic?

# Lecture 3

## Word problems

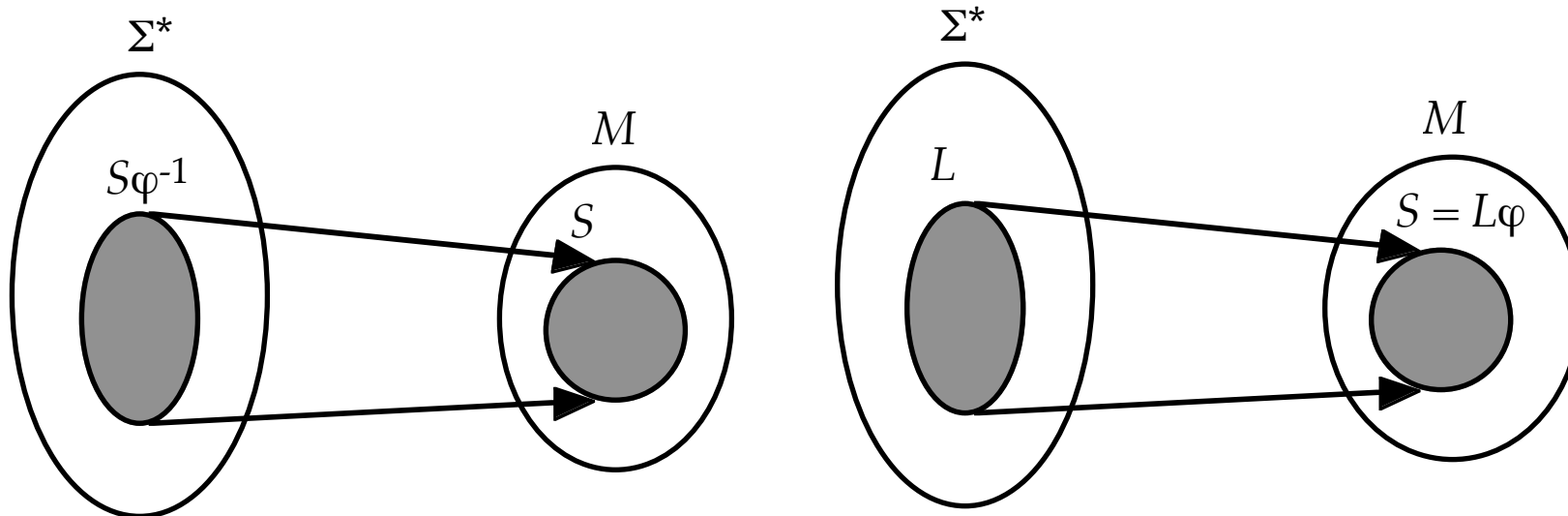
## Recall from Lecture 2 .....

$S$  is said to be *recognizable* if  $L = S\varphi^{-1}$  is a regular subset of  $\Sigma^*$ .  
We write  $S \in \mathcal{Rec}(M)$ .

$S$  is said to be *rational* if  $S = L\varphi$  for some regular subset  $L$  of  $\Sigma^*$ .  
We write  $S \in \mathcal{Rat}(M)$ .

$S$  is said to be *context-free* if  $L = S\varphi^{-1}$  is a context-free subset of  $\Sigma^*$ .  
We write  $S \in \mathcal{CF}(M)$ .

$S$  is said to be *algebraic* if  $S = L\varphi$  for some context-free subset  $L$  of  $\Sigma^*$ .  
We write  $S \in \mathcal{Alg}(M)$ .



recognizable/context-free    versus    rational/algebraic

Clearly

$$\mathcal{R}ec(M) \subseteq \mathcal{R}at(M) \quad \text{and} \quad \mathcal{C}F(M) \subseteq \mathcal{A}lg(M).$$

$$\mathcal{R}ec(M) \subseteq \mathcal{C}F(M) \quad \text{and} \quad \mathcal{R}at(M) \subseteq \mathcal{A}lg(M).$$

**Example.**

$$G = \{ \dots, x^{-3}, x^{-2}, x^{-1}, 1, x, x^2, x^3, \dots \} \cong \mathbf{Z}.$$

$$H = \{ \dots, x^{-4}, x^{-2}, 1, x^2, x^4, \dots \} \text{ - index 2 in } G.$$

$$Hx = \{ \dots, x^{-3}, x^{-1}, x, x^3, x^5, \dots \}. \quad G = H \cup Hx; \quad H \cap Hx = \emptyset.$$

$$\Sigma = \{a, A\}. \quad a\varphi = x, \quad A\varphi = x^{-1}.$$

$$H\varphi^{-1} = \{\alpha \in \Sigma^* : |\alpha|_a - |\alpha|_A \text{ is even}\} \text{ is regular; so } H \in \mathcal{Rec}\{G\}.$$

$$\{1\}\varphi^{-1} = \{\alpha \in \Sigma^* : |\alpha|_a = |\alpha|_A\} \text{ is context-free but not regular.}$$

So  $\{1\} \in \mathcal{CF}(G)$  but  $\{1\} \notin \mathcal{Rec}(G)$ .

However,  $\{1\}$  is clearly in  $\mathcal{Rat}(G)$  since  $\{1\} = \{\varepsilon\}\varphi$ .

In this particular group  $\mathcal{Alg}(G) = \mathcal{CF}(G)$ .



### Example.

$F$  - free group with generators  $x$  and  $y$ .

$$\Sigma = \{a, A, b, B\}. \quad a\varphi = x; \quad A\varphi = x^{-1}; \quad b\varphi = y; \quad B\varphi = y^{-1}.$$

Consider the context-free grammar (with starting symbol  $U$ ):

$$U \rightarrow xUx^{-1} \mid V; \quad V \rightarrow x^{-1}VVx \mid y.$$

If  $S$  is the subset of  $F$  generated by this grammar, then

$$S\varphi^{-1} \cap b^* = \{b^{2^n} : n \in \mathbf{N}\},$$

which is not context-free. So  $S \notin CF(F)$ .

However, by construction, we have that  $S \in Alg(F)$ .

# Word problems for groups

Let  $G$  be a group generated by a finite set  $X$ ; each element of  $G$  may be written as a word in the symbols  $x, x^{-1}$  (where  $x \in X$ ).

Let  $\Sigma = X \cup X^{-1}$ . As before we have the natural homomorphism  $\varphi : \Sigma^* \rightarrow G$ .

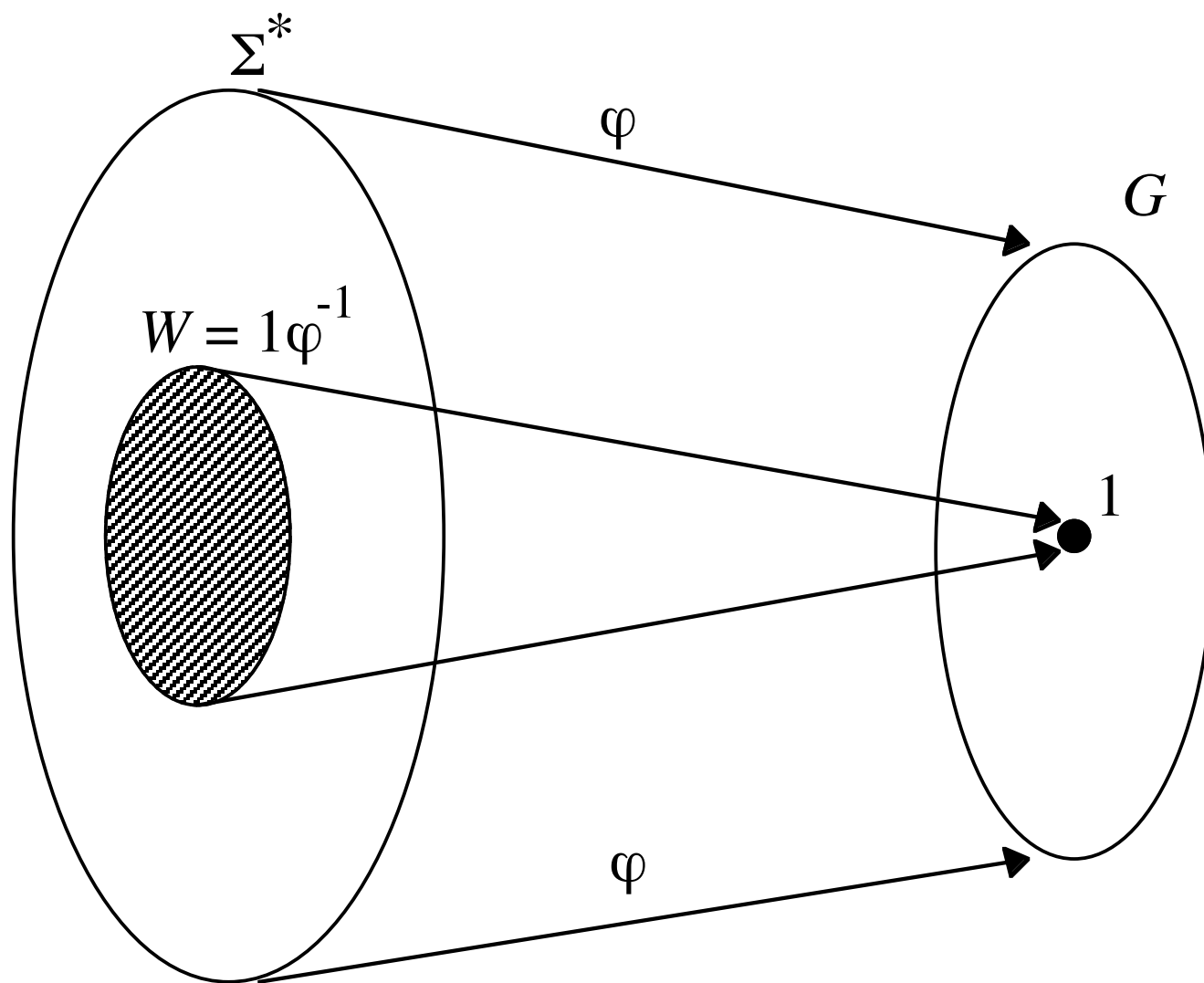
For every word  $\alpha$  in  $\Sigma^*$ , let  $[\alpha]$  denote the corresponding element of the group  $G$ .

As before, note that words  $\alpha$  and  $\beta$  represent the same element of  $G$  if and only if  $\alpha\beta^{-1}$  represents the identity element of  $G$ .

*Word problem for groups.* Given a presentation  $\langle X : R \rangle$  defining a group  $G$  and given a word  $\alpha$  in  $\Sigma^*$ , is  $[\alpha] = 1$  in  $G$ ?

Alternatively, we can define the *word problem*  $W$  for a group  $G$  generated by a finite set  $X$  to be the set of all words in  $\Sigma^*$  that represent the identity element of  $G$ .

What does the nature of the word problem (as a formal language) say about the algebraic structure of  $G$ ?



If  $G$  has a finite presentation  $\langle X : \mathcal{R} \rangle$  then  $W_X(G)$  is recursively enumerable.

The word problem for  $G$  is said to be *solvable* if  $W_X(G)$  is recursive.

If  $\mathcal{F}$  is a family of languages closed under inverse homomorphism, if  $X$  and  $Y$  are finite subsets of a group  $G$ , and if  $G = \langle X \rangle = \langle Y \rangle$ , then  $W_X(G) \in \mathcal{F}$  if and only if  $W_Y(G) \in \mathcal{F}$ .

We can just write  $W(G) \in \mathcal{F}$  in this case.

**Novikov, Boone.** There exist finitely presented groups with unsolvable word problem.

**Higman.**  $W(G)$  is recursively enumerable if and only if  $G$  is a subgroup of a finitely presented group.

**Boone & Higman.**  $W(G)$  is recursive if and only if  $G \leq H \leq K$ , where  $H$  is simple and  $K$  is finitely presented.

# Regular and context-free word problems

**Anisimov.** If  $G$  is a finitely generated group, then  $W(G)$  is regular if and only if  $G$  is finite.

**Muller & Schupp.** If  $G$  is a finitely generated group, then  $W(G)$  is context-free if and only if  $G$  has a free subgroup of finite index.

**Example.**

$G = \langle a, b : ab = ba \rangle \cong \mathbf{Z} \times \mathbf{Z}. \quad \Sigma = \{ a, A, b, B \}. \quad A = a^{-1}, B = b^{-1}.$

$W(G) = \{ \alpha \in \Sigma^* : |\alpha|_a = |\alpha|_A, |\alpha|_b = |\alpha|_B \}.$

$W(G)$  is not context-free.

However,  $\Sigma^* - W(G)$  is context-free.

**Question.** Which finitely generated groups have a word problem which is the complement of a context-free language?



Let  $\mathcal{C}$  denote the class of such (finitely generated) groups.

Some facts (Holt / Rees / Röver / Thomas).

- $\mathcal{C}$  contains all free and abelian groups.
- $\mathcal{C}$  is closed under taking finitely generated subgroups.
- $\mathcal{C}$  is closed under taking finite index overgroups.
- $\mathcal{C}$  is closed under direct products.
- there exist groups in  $\mathcal{C}$  that are not finitely presented.

**Question.** Is  $\mathcal{C}$  closed under taking free products?

# One-counter languages

*One-counter languages* - languages accepted by pushdown automata where we have only one stack symbol (apart from a bottom marker).

*Cone* - family of languages closed under

- homomorphism,
- inverse homomorphism, and
- intersection with regular languages.

**Herbst.** Let  $\mathcal{F}$  be a cone contained in the family of context-free languages and then let  $\mathcal{G}$  be the class of all finitely generated groups  $G$  such that  $W(G) \in \mathcal{F}$ . Then  $\mathcal{G}$  is the set of regular groups, one-counter groups or context-free groups.

**Herbst.** The following are equivalent for a finitely generated group  $G$ :

- $W(G)$  is one-counter.
- $G$  is either finite or has a subgroup of finite index isomorphic to  $\mathbb{Z}$ .

**Herbst.** Let  $G$  be a finitely generated group. Then the following conditions are equivalent:

- $W(G)$  is one-counter.
- $CF(G) = Alg(G)$ .
- $CF(G) = Rat(G)$ .

**Question.** Which groups satisfy  $Rat(G) = Alg(G)$  ?

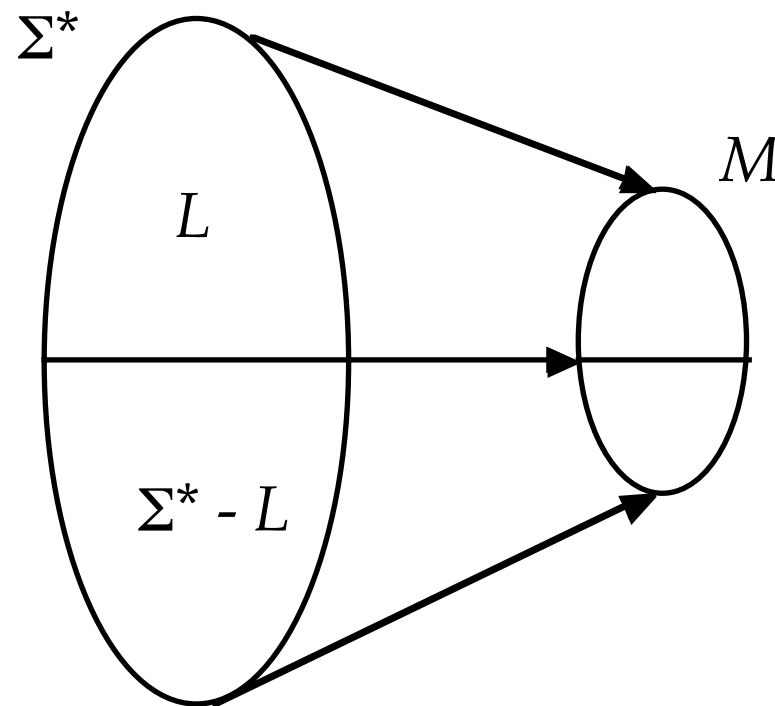
Every one-counter group has this property but other groups do as well (such as all groups with an abelian subgroup of finite index).

## Lecture 4

# Syntactic monoids and automatic groups

# Syntactic monoids

General idea of language recognition : construct a device that distinguishes elements of  $L$  from  $\Sigma^* - L$ .



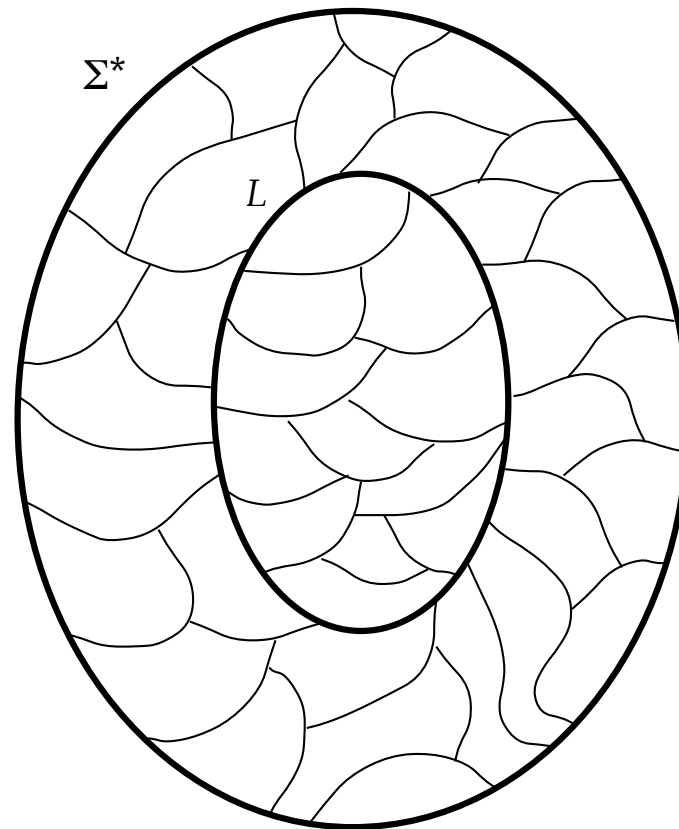
*Idea:* map  $\Sigma^*$  to another monoid  $M$  such that elements of  $L$  are distinguished (in  $M$ ) from elements of  $\Sigma^* - L$ .

The map  $\varphi$  should be a homomorphism, i.e.

$$(\alpha\beta)\varphi = (\alpha\varphi)(\beta\varphi)$$

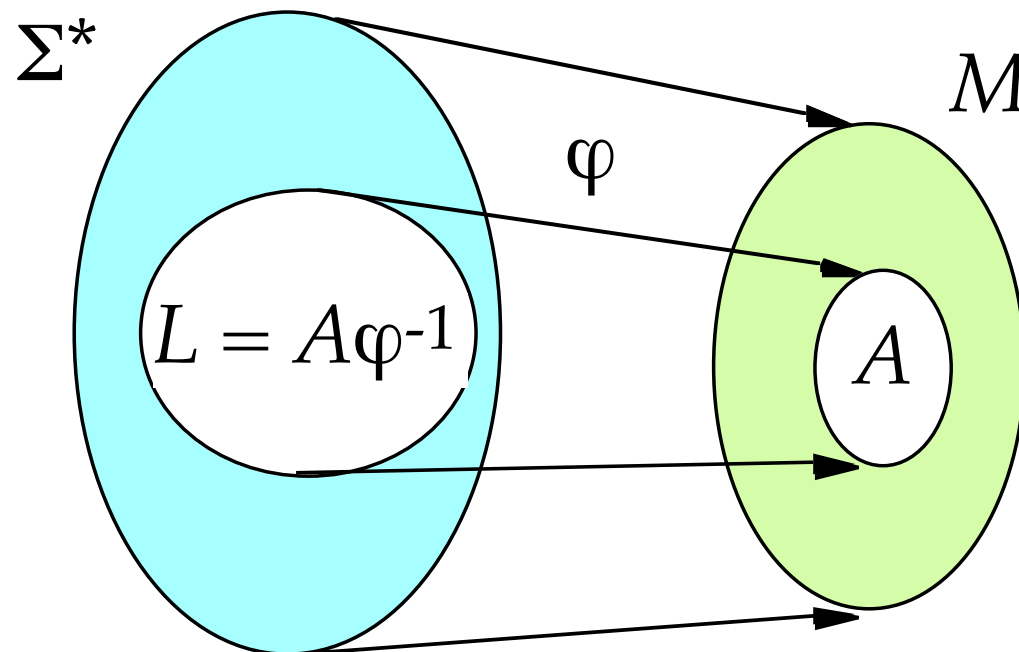
for all  $\alpha$  and  $\beta$ .

The *syntactic monoid*  $M_L$  of  $L$  is (essentially) the smallest such monoid  $M$ .



If we have a homomorphism  $\varphi$  from  $\Sigma^*$  to  $M$ , then  $M$  is isomorphic to  $\Sigma^* / \approx$  for some congruence  $\approx$ .

The *syntactic congruence*  $\approx_L$  on  $\Sigma^*$  is the coarsest congruence  $\approx$  on  $\Sigma^*$  such that  $L$  is a union of congruence classes;  $M_L$  is then  $\Sigma^* / \approx_L$ .





A subset  $A$  of a monoid  $M$  is said to be *syntactic* if there is no non-trivial congruence  $\sim$  on  $M$  such that  $A$  is a union of  $\sim$  classes.

If  $M$  is a monoid,  $\varphi : \Sigma^* \rightarrow M$  is a surjective homomorphism,  $A$  is a syntactic subset of  $M$ , and  $L = A \varphi^{-1}$ , then  $M = M_L$ .

If  $\varphi : \Sigma^* \rightarrow M$  and  $\psi : \Omega^* \rightarrow M$  are surjective homomorphisms,  $\mathcal{F}$  is a family of languages closed under inverse homomorphism and  $A \subseteq M$ , then  $A\varphi^{-1} \in \mathcal{F} \Leftrightarrow A\psi^{-1} \in \mathcal{F}$ .

$L$  is regular if and only if  $M_L$  is finite.

# Context-free languages

What about context-free languages?

regular languages  $\rightarrow$  finite monoids;

One might ask .....

context-free languages  $\rightarrow$  ?

But .....

$M_L$  is isomorphic to  $M_{\Sigma^*-L}$ ;

the class of context-free languages is not closed under  
complementation.

Every group is the syntactic monoid of its word problem.

**Parkes & Thomas.** Let  $G$  be a finitely-generated group with an element of infinite order and let  $\mathcal{F}$  be a family of languages which is closed under inverse homomorphism and intersection with regular languages.

Suppose that there exists  $K \subseteq \{a\}^*$  with  $K \notin \mathcal{F}$ ; then there exists a language  $L \notin \mathcal{F}$  such that  $G = M_L$ .

Every finite group is the syntactic monoid of a regular language.

Every finitely-generated abelian group  $G$  is isomorphic to

$$\mathbf{Z}^k \times \mathbf{Z}_{d(1)} \times \mathbf{Z}_{d(2)} \times \dots \times \mathbf{Z}_{d(r)}$$

for some  $k \geq 0$ ,  $r \geq 0$ ,  $d(i) > 1$ . We call  $k$  the *rank* of  $G$ .

**Perrot & Sakarovitch.** Suppose that  $L$  is a context-free language and that its syntactic monoid is an abelian group  $G$ .

Suppose that  $G$  has rank  $k$ . Then  $L$  is deterministic if  $k \leq 1$  and non-deterministic if  $k > 1$ .

Moreover, every finitely generated abelian group is the syntactic monoid of a context-free language.

If  $J$  and  $K$  are context-free languages such that  $M_J$  and  $M_K$  do not contain zeros, then there is a context-free language  $L$  with

$$M_L = M_J \times M_K.$$

$\mathbf{Z}$  is the syntactic monoid of a (deterministic) context-free language.

$\mathbf{N}$  is not the syntactic monoid of any context-free language.

**Conjecture (Herbst).** If  $L \subseteq \Sigma^*$  is a deterministic context-free language and  $M_L$  is a group  $G$ , then the word problem of  $G$  is deterministic context-free.

This is false (Röver).

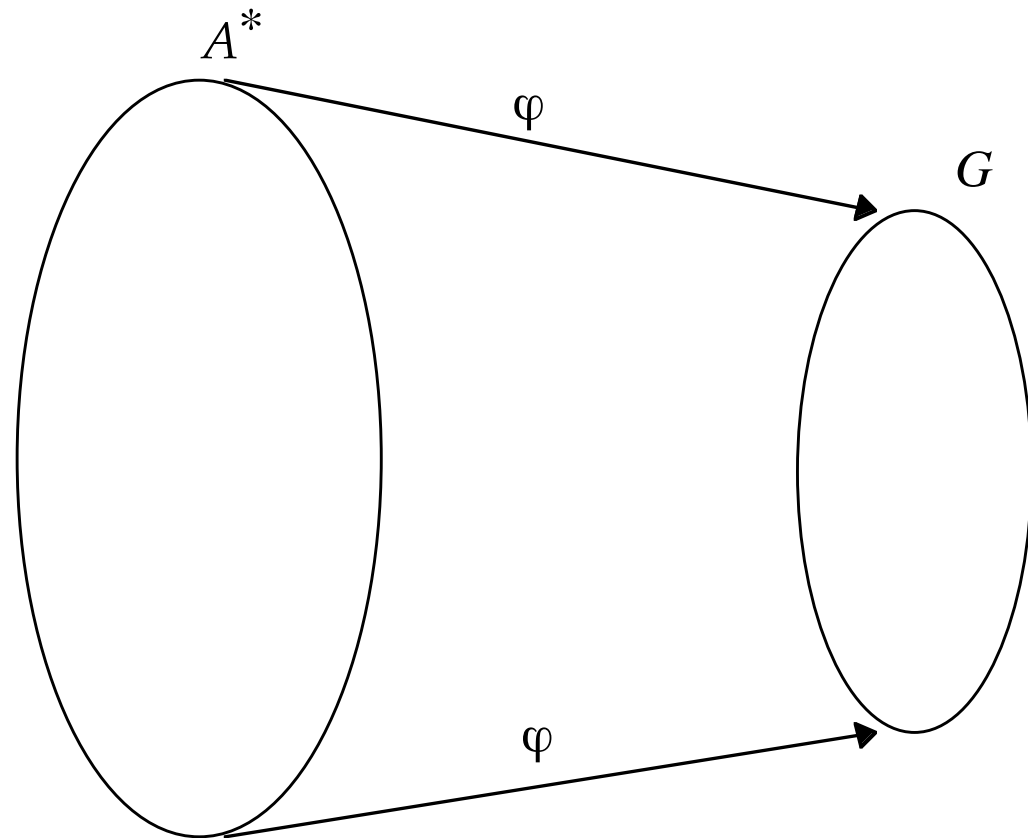
$G$  context-free word-problem  $\Rightarrow G$  the syntactic monoid of a deterministic context-free language.

$G$  co-context-free word-problem  $\Rightarrow G$  the syntactic monoid of a context-free language.

**Question.** If  $G$  is the syntactic monoid of a deterministic context-free language does  $G$  have a co-context-free word-problem ?

# Notions of automaticity

We have a group  $G$  with a (monoid) set of generators  $A$ ; we have the natural homomorphism  $\varphi : A^* \rightarrow G$ . Each element of  $G$  is represented by several words in  $A^*$ .



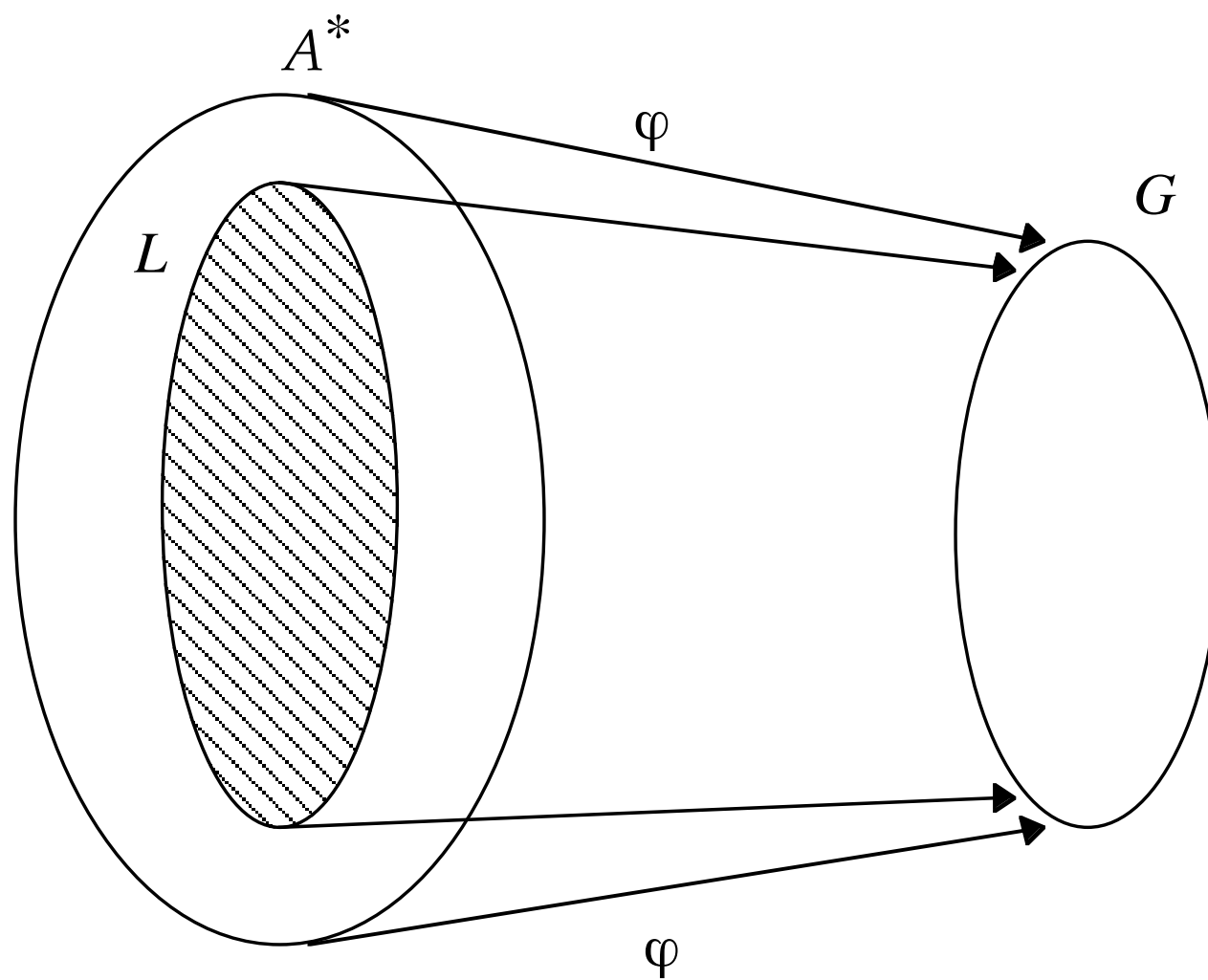
**Example.**  $G$  is the group  $\langle a, b : ab = ba \rangle$  and  $A = \{ a, b, a^{-1}, b^{-1} \}$ .

The words  $a, abb^{-1}, aa^{-1}a, b^{-1}ab, b^2ab^{-2}, \dots$  all represent the same element of  $G$ .

We could consider the set of words of the form  $a^i b^j$  with  $i, j \in \mathbf{Z}$  (where we interpret  $a^{-n}$  as  $(a^{-1})^n$  for  $n > 0$  and similarly for  $b^{-n}$ ); this forms a regular language which maps onto  $G$ .

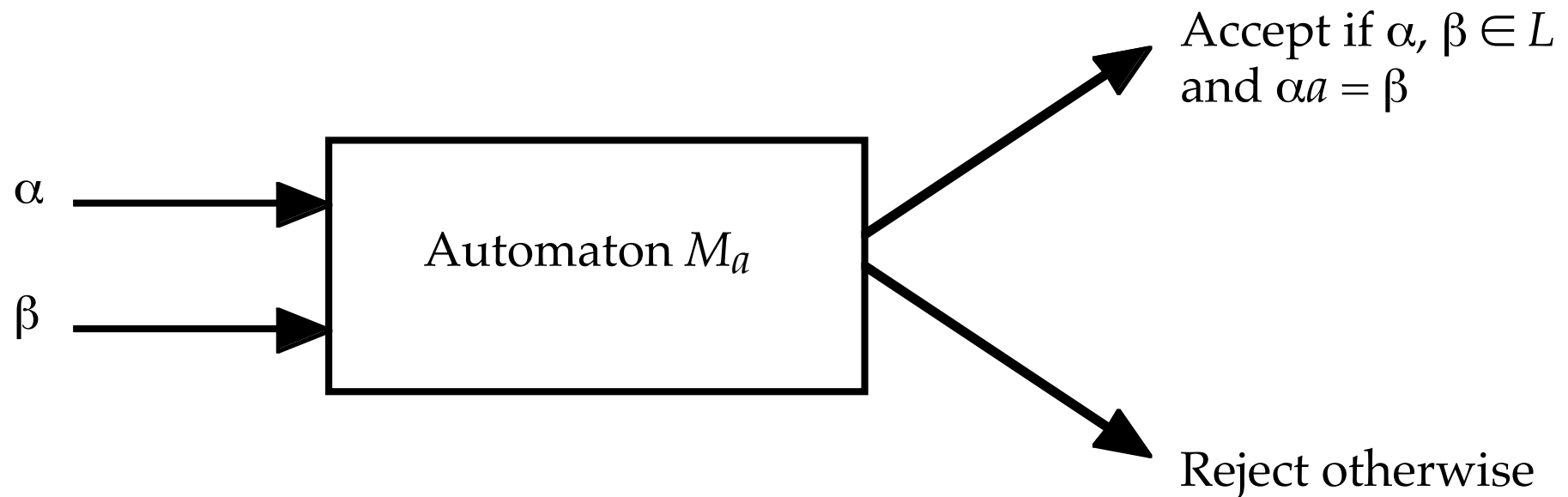
We will assume we have a regular language  $L \subseteq A^*$  such that  $L$  maps onto  $G$ .






# Automatic groups (Epstein et al)

For each  $a \in A \cup \{\varepsilon\}$  there is a finite automaton  $M_a$  such that




We can either read our words synchronously (we say that the group is *automatic*) .....

$a_1$	$a_2$	$a_3$	.....	$a_n$	\$	.....	\$
$b_1$	$b_2$	$b_3$	.....	$b_n$	$b_{n+1}$	.....	$b_m$




..... or asynchronously (we say that the group is *asynchronously automatic*):

$a_1$	$a_2$	$a_3$	.....	$a_n$
-------	-------	-------	-------	-------



$b_1$	$b_2$	$b_3$	.....	$b_n$	$b_{n+1}$	.....	$b_m$
-------	-------	-------	-------	-------	-----------	-------	-------



We say that  $(A, L)$  is an *automatic structure* for  $G$ .

We can assume that  $L$  maps bijectively to  $G$ ;  $(A, L)$  is then an automatic structure *with uniqueness*.

Some advantages of this notion.

1. Captures a wide class of groups.
2. Some computation is effective – e.g. we can solve the word problem in quadratic time and finiteness is decidable.

All of this (including the uniqueness) generalizes naturally to semigroups/monoids.

D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. Levy, M. S. Paterson and W. Thurston, “Word processing in groups”.

**Epstein et al.** If  $G$  is an automatic group, then we can solve the word problem for  $G$  in quadratic time.

**Campbell, Robertson, Ruskuc & Thomas.** This generalizes to monoids.

Not everything generalizes to monoids ...

**Epstein et al.** Automatic groups are finitely presented.

**Example.** This does not hold for automatic monoids.

Let  $M$  be the monoid defined by the presentation:

$$\langle a, b : ab^n a = aba \text{ for } n \in \mathbf{N} \rangle.$$

$M$  is not finitely presented.

Let  $A = \{a, b\}$  and  $L$  be the regular language  $\{b\}^* \{a, ab\}^* \{b\}^*$ . Then  $(A, L)$  is an automatic structure for  $M$  (with uniqueness).

A language  $L$  is said to be *prefix-closed* if, whenever  $\alpha \in L$  and  $\beta$  is a prefix of  $\alpha$ , then  $\beta \in L$ .

An automatic structure  $(A, L)$  is said to be *prefix-closed* if  $L$  is prefix-closed.

**Epstein et al.** Every automatic group has a prefix-closed automatic structure.

**Question.** Does every automatic monoid have a prefix-closed automatic structure?

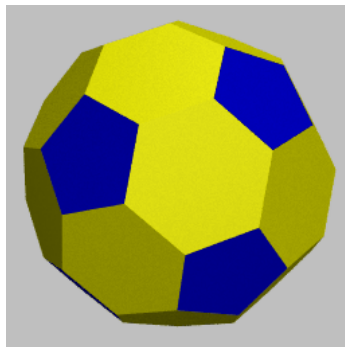
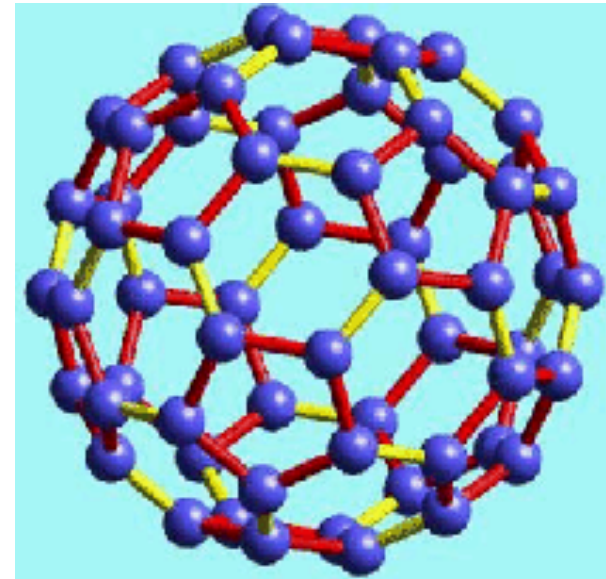
## Fellow traveller property

If  $G$  is a group generated by a finite set  $X$ , and if  $\Sigma$  is the alphabet  $X \cup X^{-1}$ , then we define the *Cayley graph* of  $G$  with respect to  $X$  to be the graph whose vertices are the elements of  $G$  and such that, for each  $x \in \Sigma$  and  $g \in G$ , there is a directed edge (labelled  $x$ ) from  $g$  to  $gx$ .

We often identify traversing an edge corresponding to  $x^{-1}$  from  $gx$  to  $g$  with traversing the edge corresponding to  $x$  from  $g$  to  $gx$  in the opposite direction.

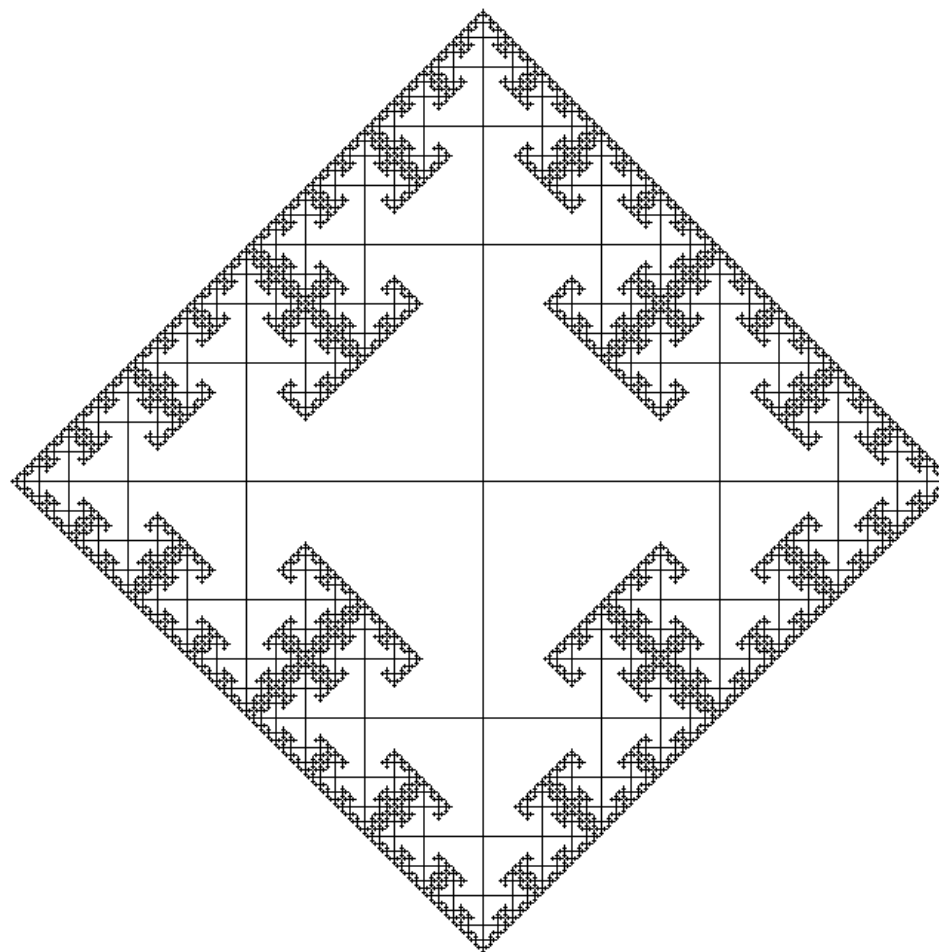


For a finite group  $G$ , we obviously get a finite graph; for example, the Cayley graph of the group  $A_5$  of all even permutations of a set of five objects is shown on the right.

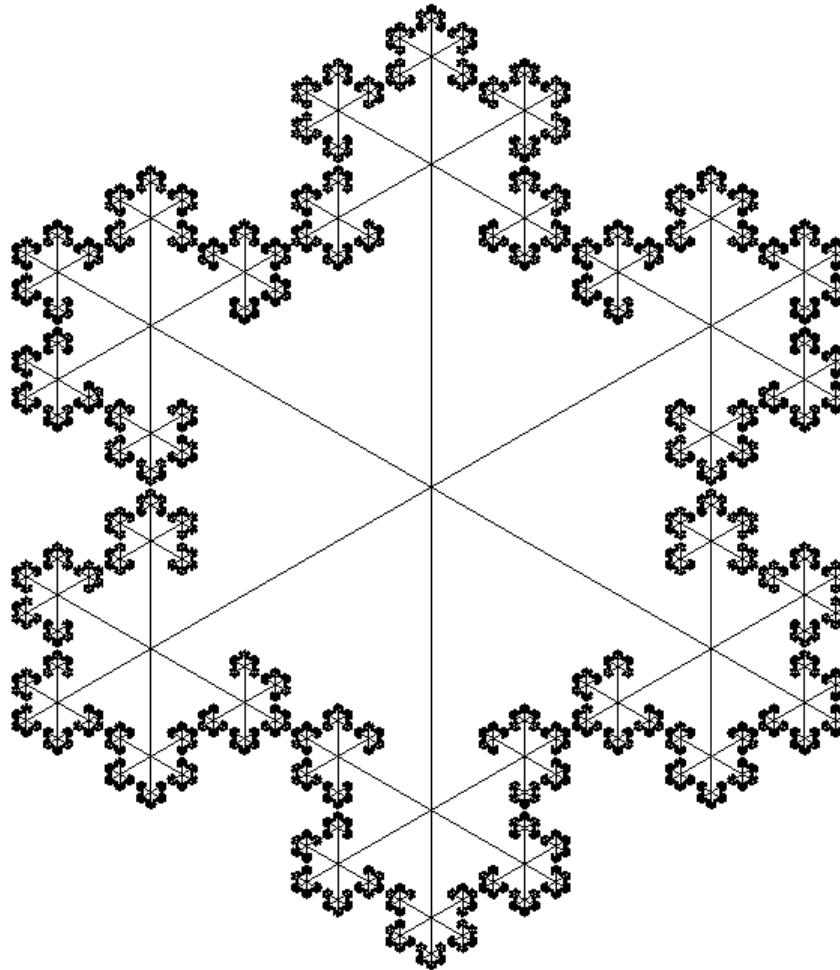


This Cayley graph occurs in many common objects!

For an infinite group we have an infinite Cayley graph; on the right is a representation of the Cayley graph of the free group of rank 2.

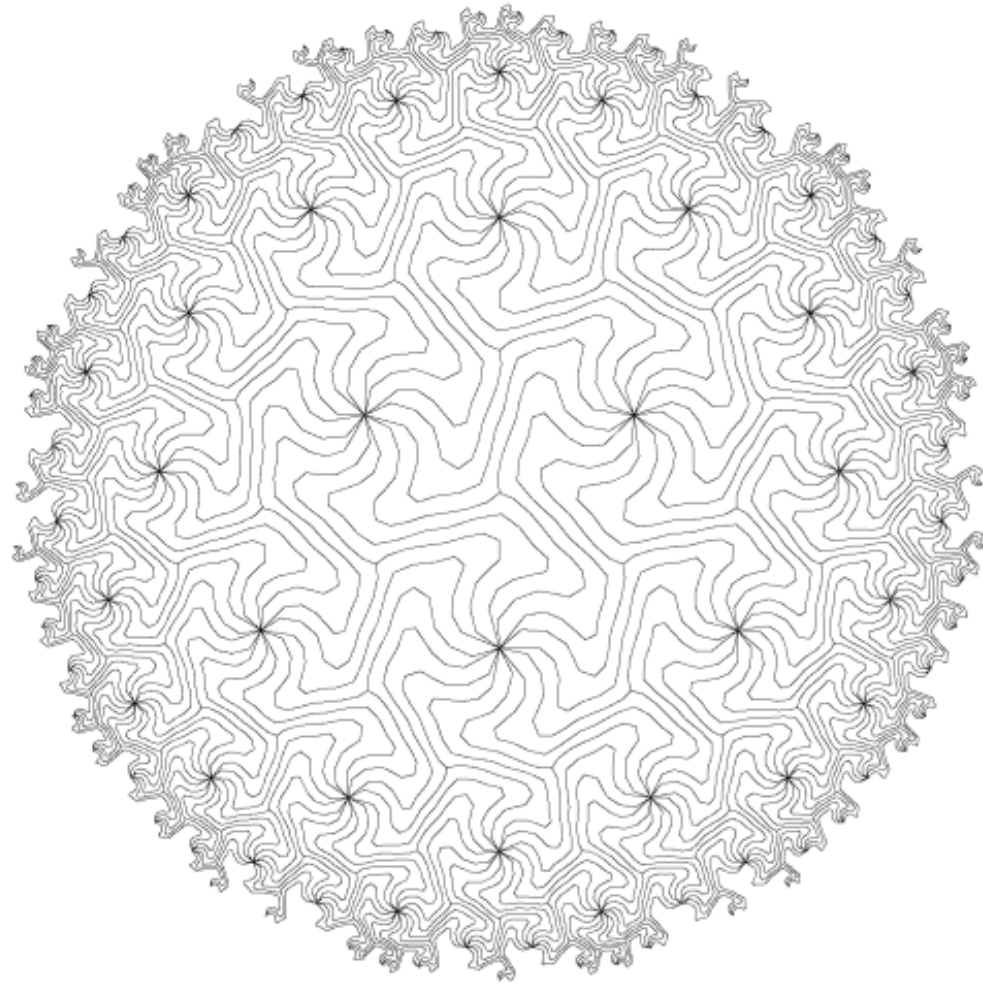


Here is a representation of the Cayley graph of the free group of rank 3.

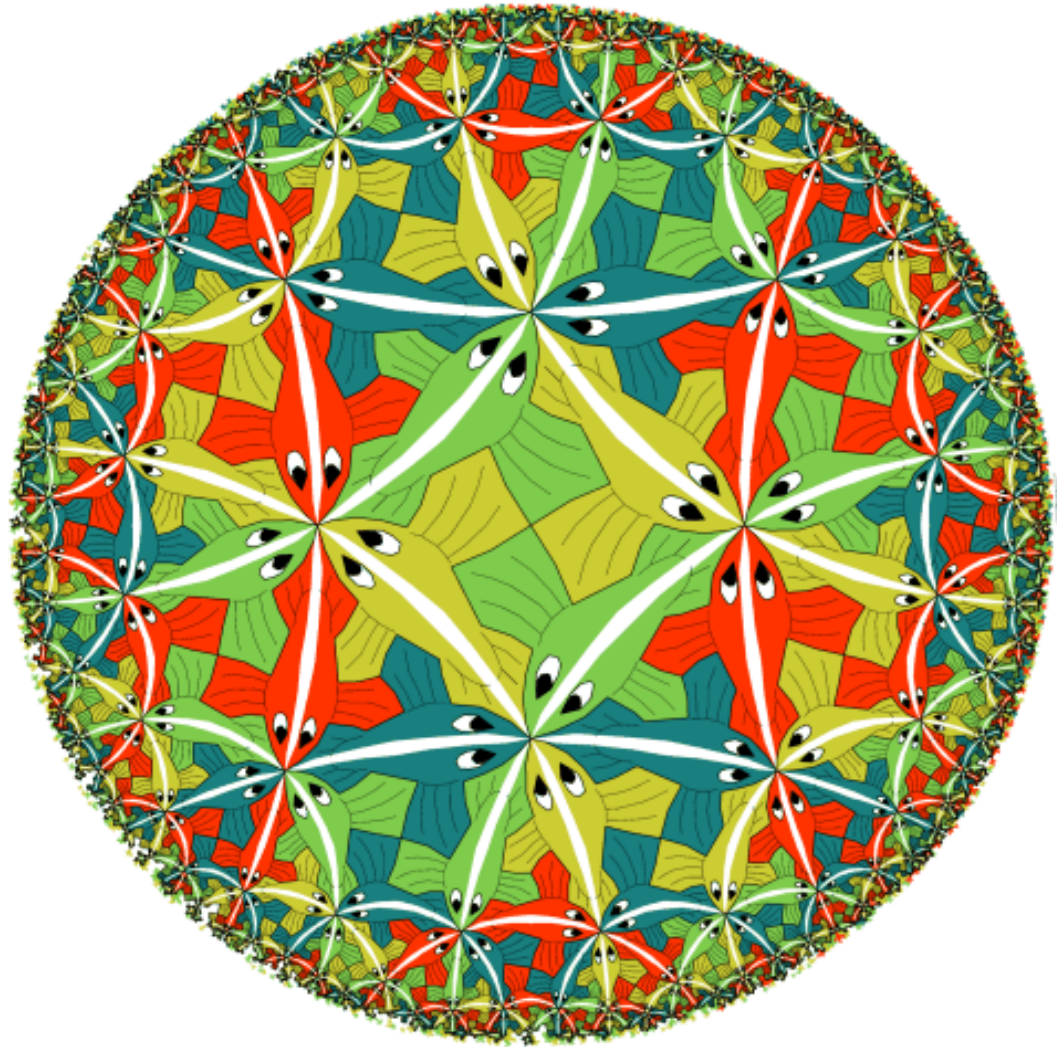


Here is a representation of the Cayley graph of the group with generators  $x$  and  $y$  and relations

$$x^2 = y^3 = (xy)^7 = 1.$$



Such Cayley graphs gives a natural tiling of hyperbolic space; this is used by artists such as Escher.



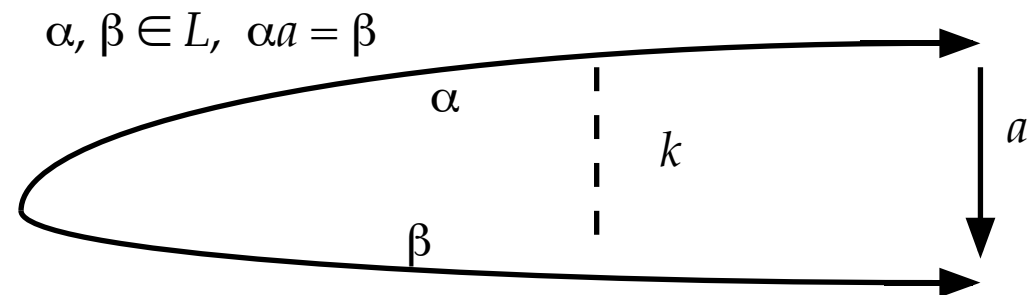
Note that a word  $\alpha$  in  $\Sigma^*$  is in  $W_X(G)$  if and only if  $\alpha$  corresponds to a closed path in the Cayley graph  $\Gamma$ .

Let  $G$  be a group with Cayley graph  $\Gamma$ .

There is a natural distance function  $d$  on  $\Gamma$ ; if  $x$  and  $y$  are vertices, then  $d(x, y)$  is the shortest length of a path from  $x$  to  $y$ .

The concept generalizes naturally to monoids.

In groups, we have a nice geometric characterization of automaticity (the *fellow traveller property*).



The situation in automatic monoids is much more complicated but there is a complete geometric characterization (Hoffmann & Thomas).