The word problem for semigroups

Rick Thomas

Department of Computer Science

University of Leicester

NBSAN, York

23rd November 2011

Generating sets

If Σ is a finite set of symbols, then we let Σ^* denote the set of all finite words of symbols from Σ (including the empty word ϵ). If we only want to consider non-empty words, then we denote the resulting set by Σ^+.

Σ^+ is the *free semigroup* on Σ and Σ^* is the *free monoid* on Σ.

If we have a group G (or a monoid M) with a finite set of generators Σ, then we have a natural homomorphism $\varphi : \Sigma^* \to \mathrm{G}$ (or $\varphi : \Sigma^* \to \mathrm{M}$).

For a semigroup S generated by a finite set Σ we have a natural homomorphism $\varphi : \Sigma^+ \to \mathrm{S}$.

Word problems

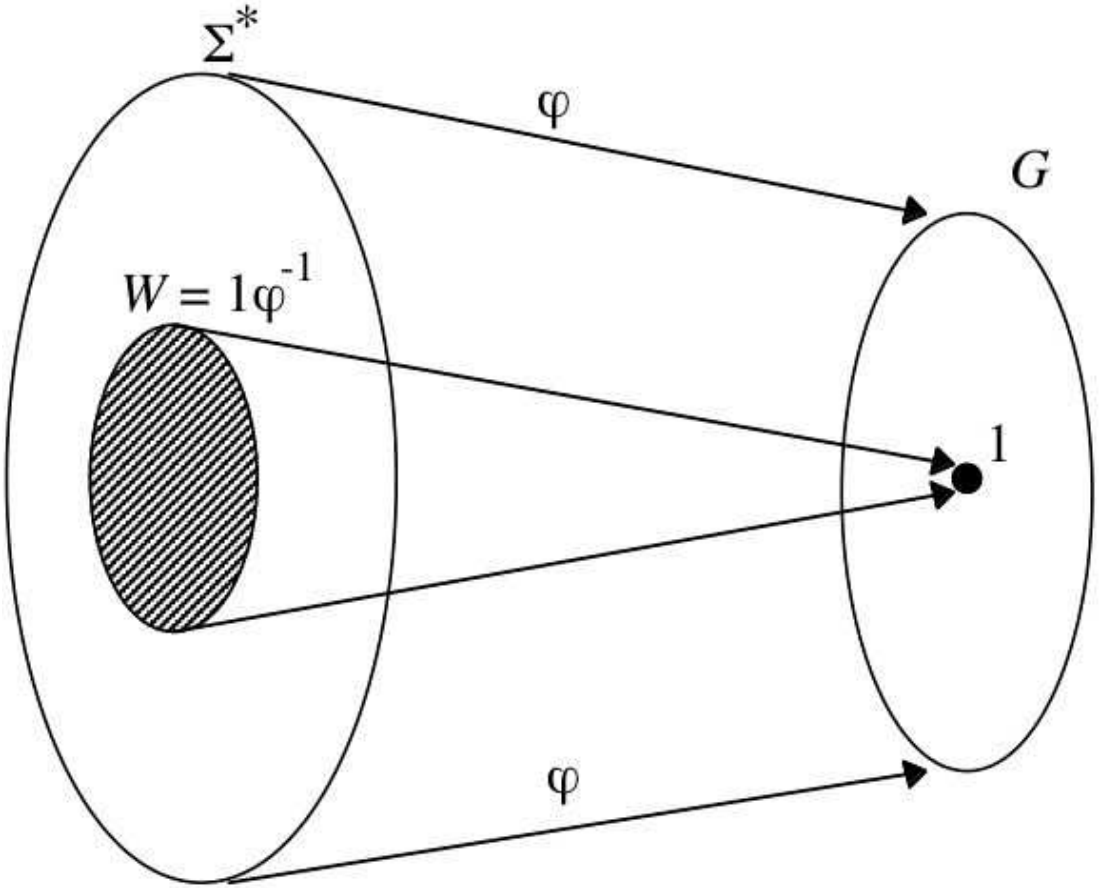
The *word problem* in such a structure is the following question:

Input: Two words α and β in Σ^* (or Σ^+ in the case of a semigroup);

Output: **Yes** if α and β represent the same element of the group (monoid, semigroup);

No otherwise.

In a group, given a word β representing an element g, let γ be a word representing g^{-1}. Now α and β represent the same element of the group if and only if $\alpha\gamma$ represents the identity.


Σ*
φ
G
W = 1φ-1
1
φ

Word problems

Given this, we can define the word problem $W = W(\mathrm{G})$ of a group G to be the set of all words in Σ^* that represent the identity element of G.

(This is not appropriate for monoids and does not make sense in semigroups.)

In this way, we can think of the word problem of a group as being a formal language.

We will focus on some relatively simple classes of languages, the *regular languages*, the *linear languages* and the *context-free languages*. Saying that the word problem of a group G is regular (or linear or context-free) does not depend on the choice of finite generating set for G.

Regular grammars

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

We have a finite set Σ of *terminals* that cannot be rewritten.

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

Each production rule is of the form $\mathrm{A} \to \mathrm{x}\mathrm{B}$, where $\mathrm{A}, \mathrm{B} \in \mathrm{N}$ and $\mathrm{x} \in \Sigma$, or else of the form $\mathrm{A} \to \epsilon$.

There is a designated starting symbol $S \in \mathrm{N}$. The *language generated by* the grammar is the set of all words in Σ^* that can be derived from S.

A language is said to be *regular* if there is a regular grammar generating it.

Example

$G = (N, \Sigma, P, S)$.

$N = \{S, T\}; \qquad \Sigma = \{a, b, c\}$.

P is the set of productions:

$$S \rightarrow aS \mid bT, \;\; T \rightarrow cT \mid \epsilon.$$

This regular grammar G generates the language

$$\{a^n bc^m : n, m \in \mathbb{N}\}.$$

Other grammars

We have various generalizations of the notion of a regular grammar:

linear grammar: allows rules of the form $A \to xB$ and $A \to Bx$ (with $A, B \in N$ and $x \in \Sigma$) as well as $A \to \epsilon$;

one can also allow rules of the form $A \to x$ with $A \in N$ and $x \in \Sigma$ (as this is equivalent to $A \to xB$ and $B \to \epsilon$ for some new $B \in N$);

context-free grammar: allows any rule of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$.

These types of grammar generate the classes of *linear languages* and *context-free languages* respectively.

Examples

$G_1 = (N_1, \Sigma_1, P_1, S)$.

$N_1 = \{S, T\}$.

$\Sigma_1 = \{a, b, c\}$.

P_1 is the set of productions:

$$S \to aT \mid b, \quad T \to Sc.$$

The linear grammar G_1 generates the language

$$\{a^n b c^n : n \in \mathbb{N}\}.$$

$G_2 = (N_2, \Sigma_2, P_2, X)$.

$N = \{X\}$; $\Sigma_2 = \{a, \overline{a}, b, \overline{b}\}$.

P_2 is the set of productions:

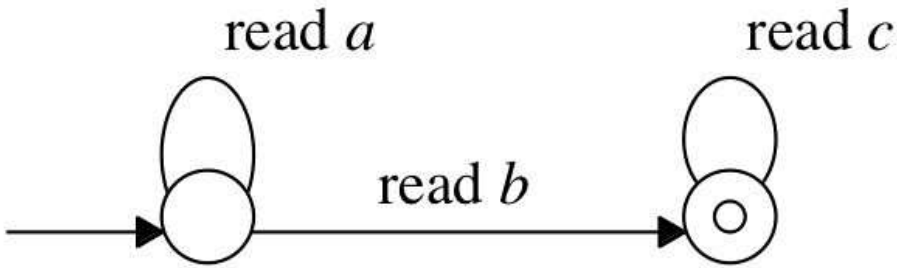
$$X \to aX\overline{a} \mid \overline{a}Xa \mid bX\overline{b} \mid \overline{b}Xb \mid XX \mid \epsilon.$$

The context-free grammar G_2 generates the word problem of the free group of rank 2 on $\{a, b\}$ (where $\overline{a}$ represents a^{-1} and $\overline{b}$ represents b^{-1}).

Automata

We can also define these classes of languages using various notions of "automata".

Regular languages are accepted by *finite automata*.

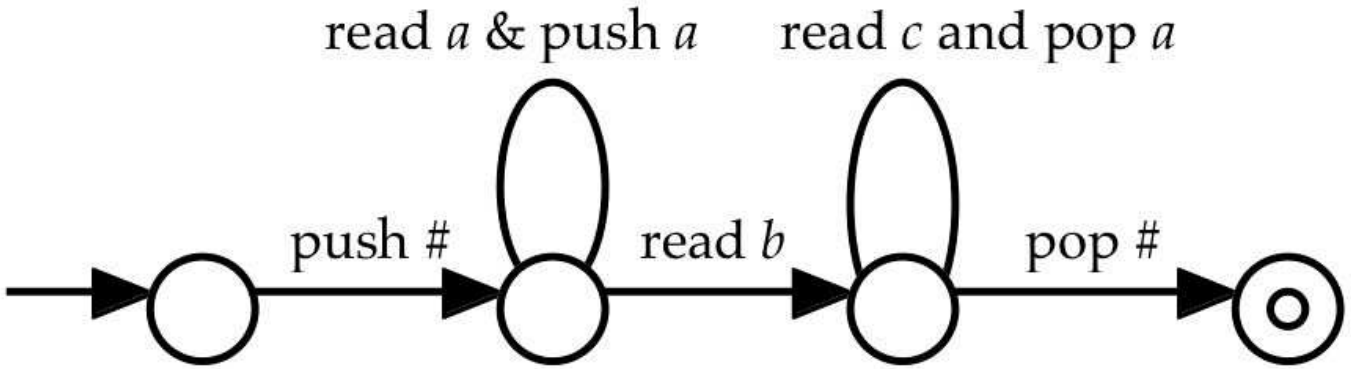


A word is *accepted* if we reach an "accept state" after reading the word.

The *language* $\mathrm{L}(\mathrm{M})$ of M is the set of all words accepted by M. The above automaton accepts $\{\mathrm{a}^{\mathrm{n}}\mathrm{bc}^{\mathrm{m}} : \mathrm{n}, \mathrm{m} \in \mathbb{N}\}$.

Pushdown automata

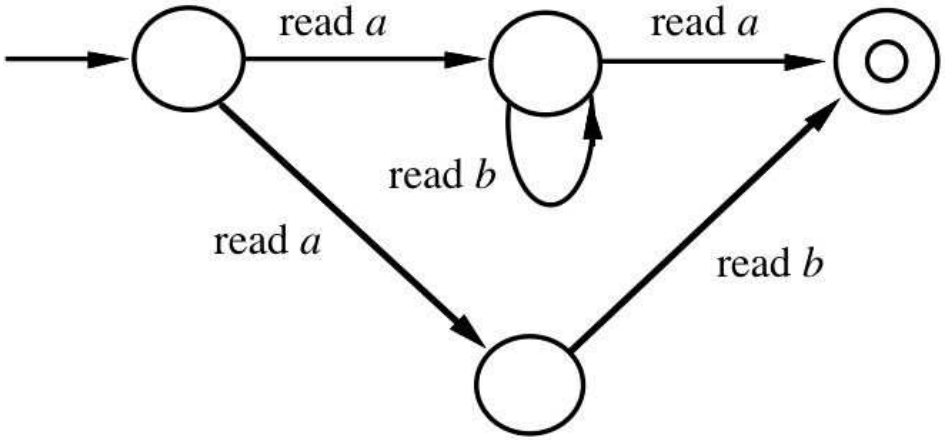
Context-free languages are accepted by *pushdown automata* where we add a “stack” to the machine.



restrict to “one turn”:	*linear languages*
restrict to one stack symbol (apart from a fixed bottom marker #):	*one-counter languages*

Determinism

An automaton is said to be *deterministic* if there can never be a possibility of choice as regards to which move to make. For example, the finite automaton shown below is not deterministic.



In a nondeterministic machine, a word is said to be *accepted* if it is *possible* to reach an accepting configuration. The automaton shown above accepts $\{ab^n a : n \in \mathbb{N}\} \cup \{ab\}$.

Word problems of groups

If G is a finitely-generated group, then $W(\mathrm{G})$ is regular if and only if G is finite. (Anisimov)

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

As a consequence, if $W(\mathrm{G})$ is context-free, then it is deterministic context-free.

If G is a finitely-generated group, then $W(\mathrm{G})$ is a one-counter language if and only if G has a cyclic subgroup of finite index. (Herbst)

(A group with a linear word problem is finite.)

Decidability

There is no algorithm that, given a context-free language L, will decide whether or not L is the word problem of a group. (Lakin & Thomas)

This can be generalized to the fact that there is no algorithm that, given a one-counter language L, will decide whether or not L is the word problem of a group. (Jones & Thomas)

However, there is an algorithm that, given a deterministic context-free language L, will decide whether or not L is the word problem of a group. (Jones & Thomas)

Word problems of semigroups

Duncan and Gilman proposed the following definition of the word problem for a semigroup S generated by a finite set A:

$$W(S) = \{\alpha\#\beta^{rev} : \alpha, \beta \in A^+, \alpha =_S \beta\}.$$

This is a natural generalization of the word problem of a group G which was

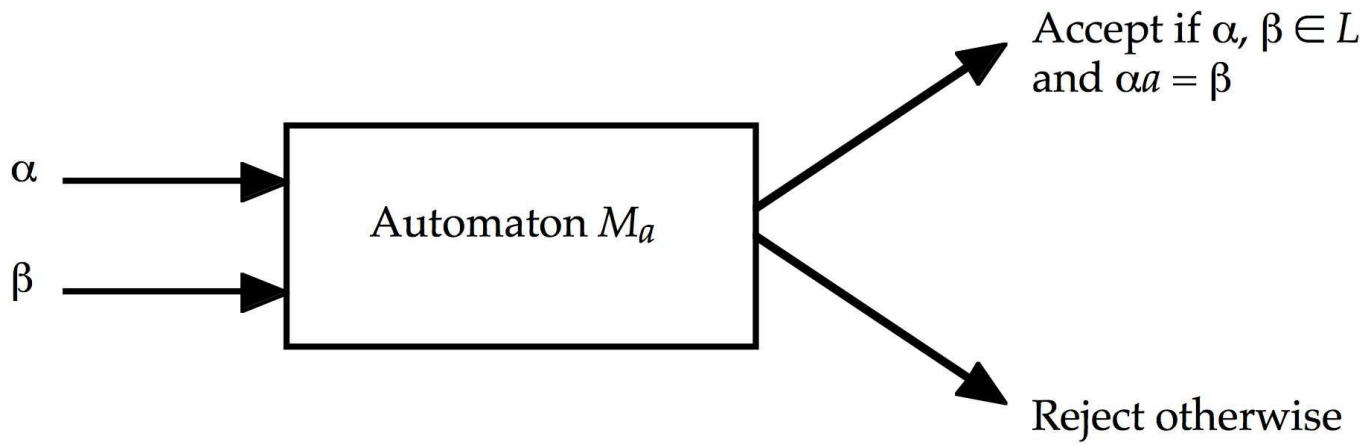
$$W(G) = \{\alpha\beta^{-1} : \alpha, \beta \in A^*, \alpha =_G \beta\}.$$

In this way, we can consider the word problem of a semigroup as a formal language.

If S is a finitely-generated semigroup, then $W(S)$ is regular if and only if S is finite. (Duncan & Gilman)

Automaticity - idea

We have a finite generating set A for a group G and a regular subset L of A^* that maps onto G. We want to perform/check the multiplication of elements by generators via finite automata.

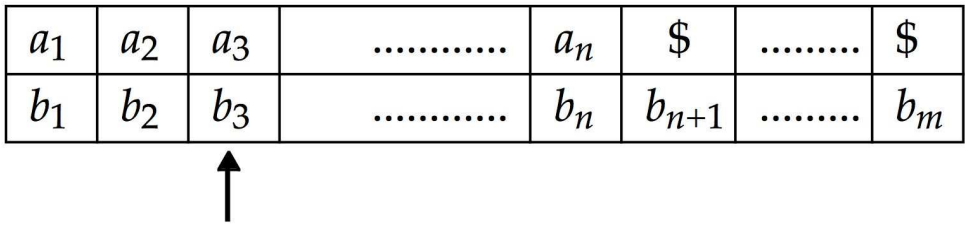


We will have one such automaton M_a for each element of $A \cup \{\epsilon\}$.

Automaticity

Our automata need to read two symbols at once (if we are considering the synchronous case).

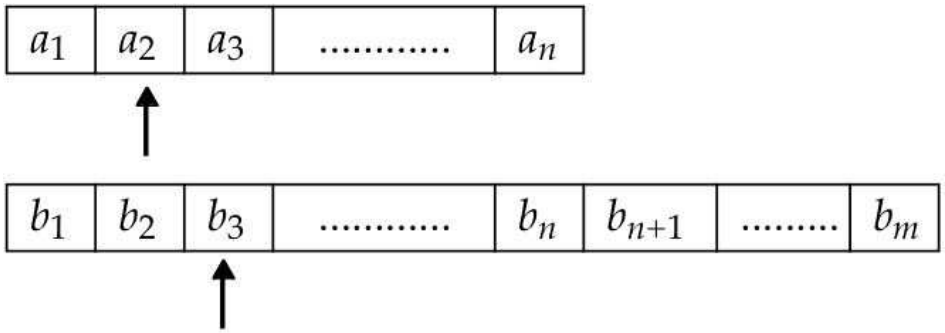
If one string is shorter than the other, then we will need to “pad” the shorter string with some extra symbols ($ say) to make the strings the same length.



This gives rise to the notion of an *automatic structure* (A, L) for G.

Asynchronous automaticity

In the asynchronous case, we can read from the two tapes at different speeds:



This gives rise to the notion of an *asynchronous automatic structure* (A, L) for a group G.

Automatic semigroups

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

A semigroup S is automatic if and only if S^1 is automatic. (Campbell, Robertson, Ruškuc & Thomas)

In semigroups (unlike groups) one has to be careful about which side you put the paddings and which side you take the multiplication by generators.

Hyperbolicity

There are many definitions of a *hyperbolic group*; an elegant approach (due to Gilman) generalizes naturally to monoids (and semigroups).

We have a monoid M generated by a finite set A. We then have a regular language $\mathrm{L} \subseteq \mathrm{A}^*$ mapping onto the monoid M; the *multiplication* of elements of L is then *checked* by a pushdown automaton.

To be more precise, we insist that the language

$$\{\alpha\#\beta\#\gamma^{rev} : \alpha, \beta, \gamma \in \mathrm{L}, \alpha\beta =_{\mathrm{M}} \gamma\}$$

is context-free. (Duncan & Gilman)

This is equivalent to the normal definitions of hyperbolic if we restrict ourselves to groups.

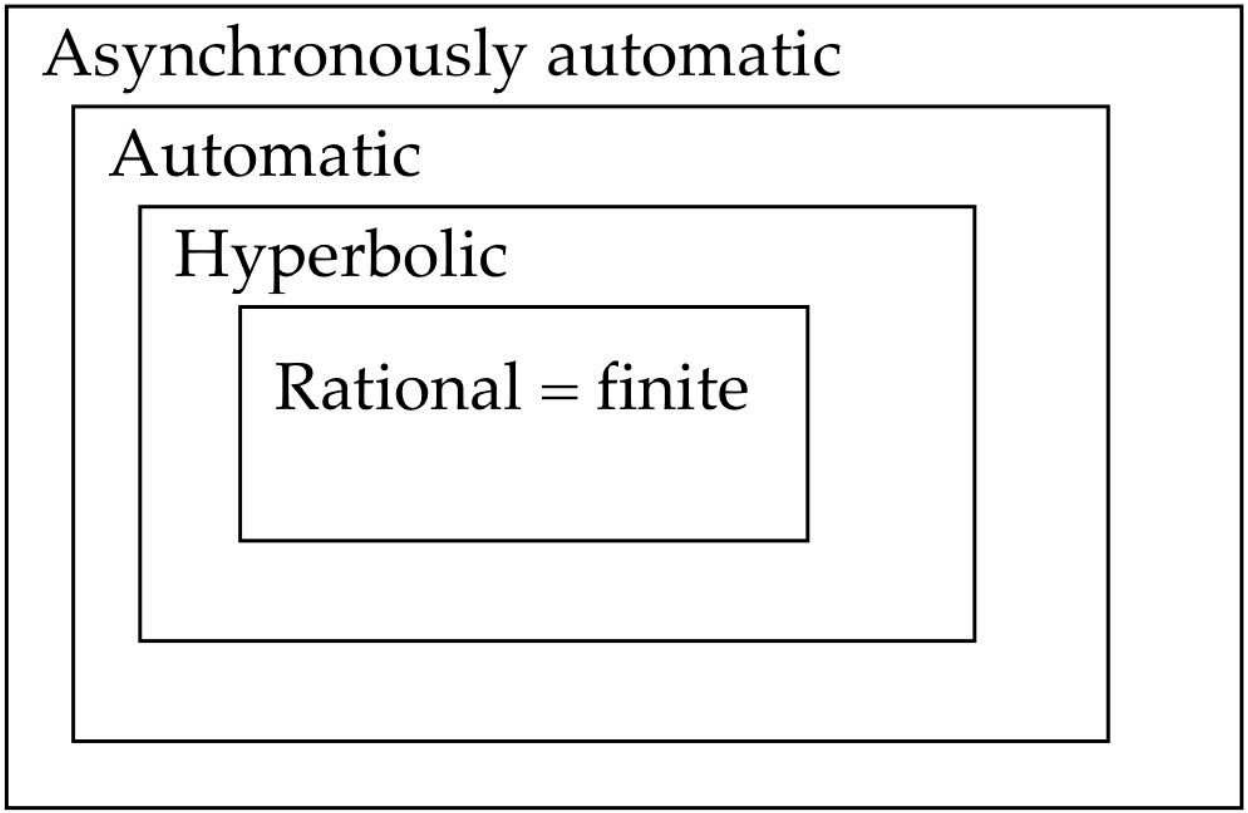
Rational monoids and semigroups

This notion is due to Sakarovitch.

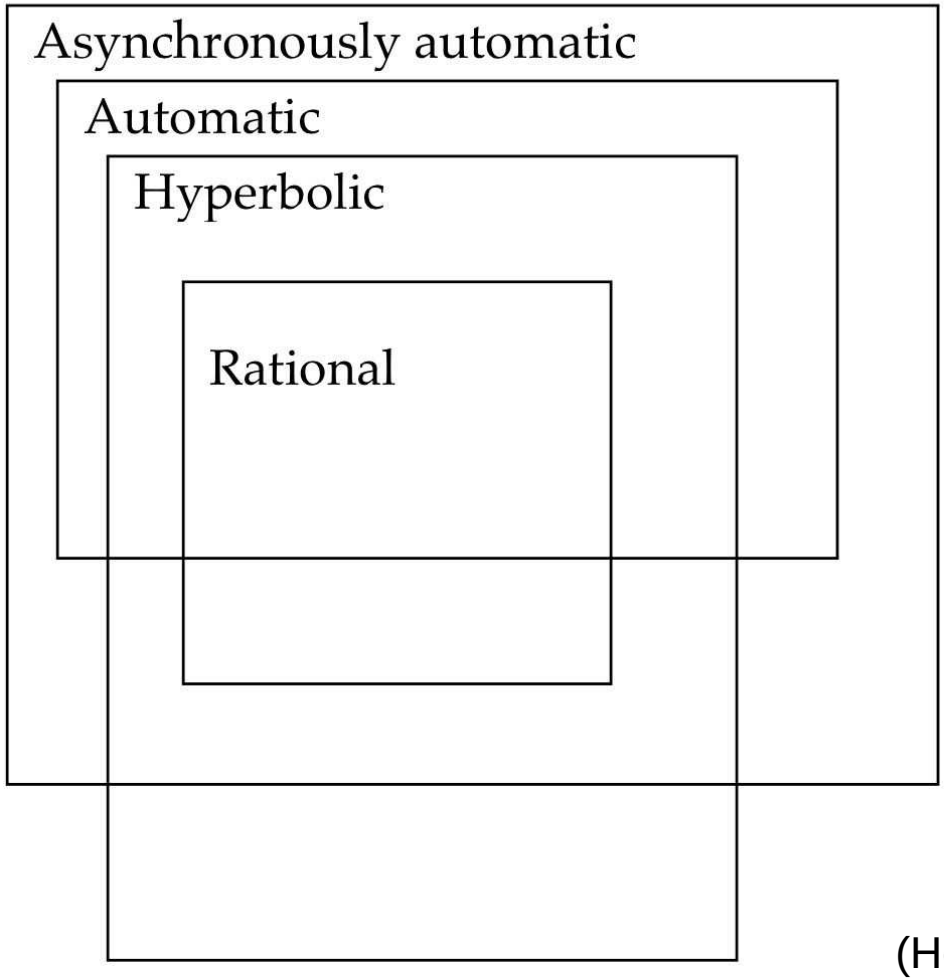
We have a monoid M (or semigroup S) generated by a finite set A. We then have a regular language $L \subseteq A^*$ mapping bijectively onto M and a transducer such that

A group is rational if and only if it is finite. (Sakarovitch)

Groups


Asynchronously automatic
Automatic
Hyperbolic
Rational = finite

Semigroups



(Hoffmann, Kuske, Otto & Thomas)

Linear and one-counter word problems

A semigroup has a linear word problem if and only if it is rational.
(Hoffmann, Holt, Owens & Thomas)

If a finitely generated semigroup S has word problem a one-counter language, then S has a linear growth function. (Holt, Owens & Thomas)

If S is a finitely generated semigroup with linear growth then there exist finitely many elements $a_i,\ b_i,\ c_i \in S \cup \{\epsilon\}$ such that every element of S is represented by a word of the form $a_i b_i^n c_i$ for some i and some $n \geqslant 0$.
(Holt, Owens & Thomas)

Context-free word problems

Some partial results: (Hoffmann, Holt, Owens & Thomas)

- there exists a semigroup with a word problem that is context-free but not deterministic context-free;
- if S is a finitely generated semigroup and T has finite Rees index in S, then S has context-free word problem if and only if T has context-free word problem;
- if S and T have context-free word problems, then the free product $S * T$ has context-free word problem;
- if S has a context-free word problem then S is hyperbolic (but need not be automatic).

Thank you!