Word problems of groups, formal languages and decidability

Rick Thomas

University of Leicester

Department of Computer Science

http://www.cs.le.ac.uk/people/rmt/

Languages

Σ : finite set of symbols.

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

Σ^+ forms a semigroup (under the operation of concatenation).

Σ^* : the set of all finite words formed from the symbols in Σ (including the *empty word* ε).

Σ^* forms a monoid (under the operation of concatenation).

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

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

Some classes of languages and the associated “machines”:

<table>
<tr><th>Languages</th><th>Machines</th></tr>
<tr><td>Regular languages</td><td>Finite automata</td></tr>
<tr><td>Context-free languages</td><td>Pushdown automata</td></tr>
<tr><td>Recursive languages</td><td rowspan="2">Turing machines</td></tr>
<tr><td>Recursively enumerable languages</td></tr>
</table>

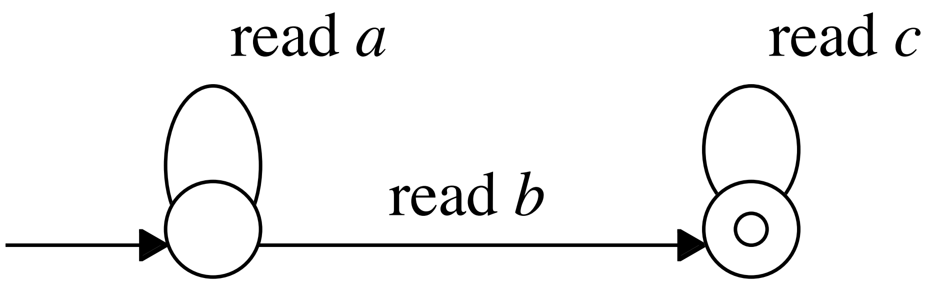
Regular languages.

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

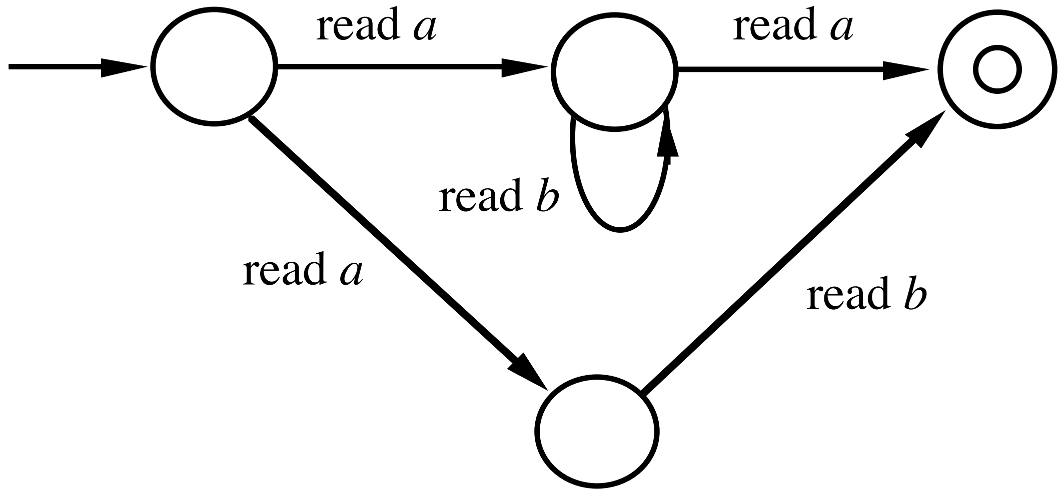
Finite automata have states with a designated *start state* and a set of *accept states*. A word α is *accepted* by an automaton if α maps the start state to an accept state.

For example, the finite automaton below accepts the language

$\{a^n b c^m : n, m \geq 0\}$:



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 $\{ab^na : n \geq 0\} \cup \{ ab \}$.

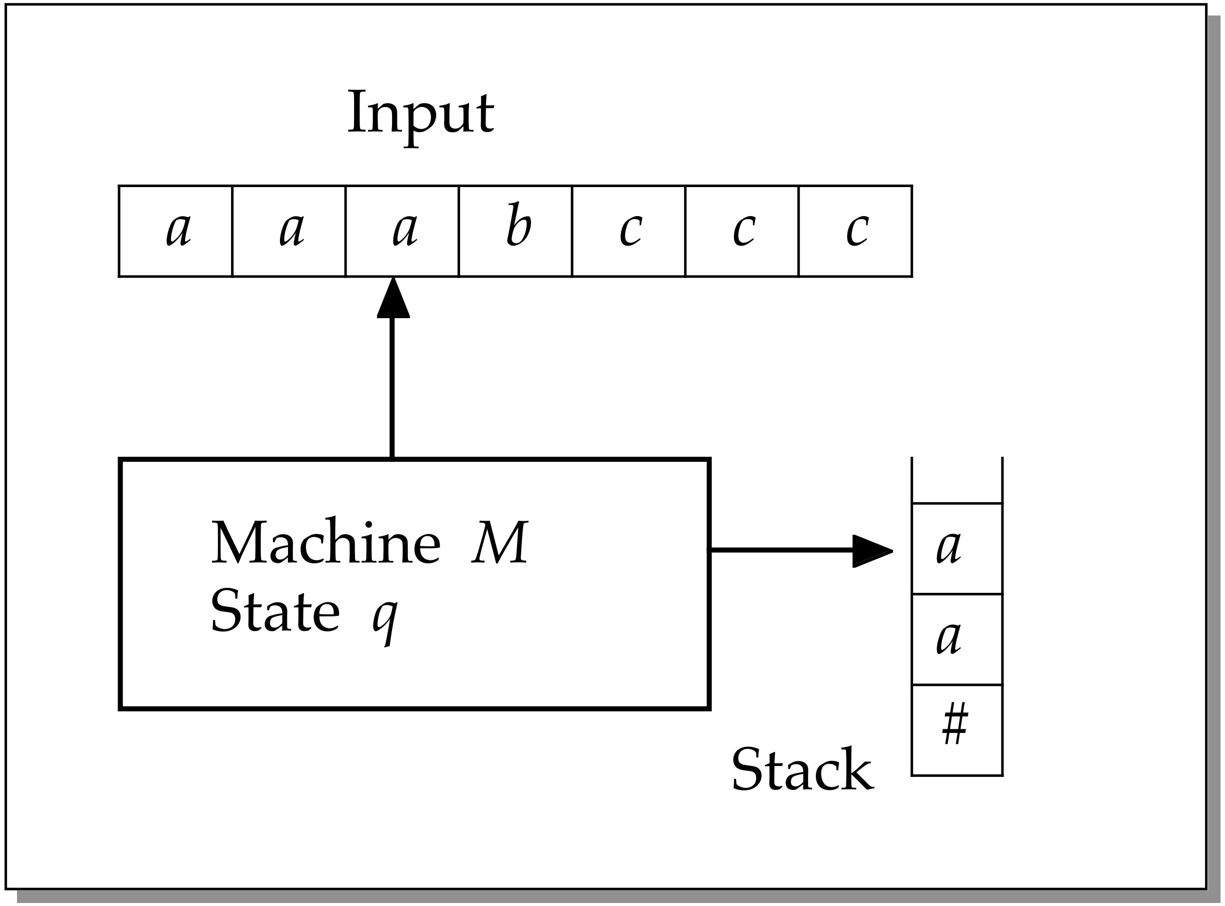
Context-free languages

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

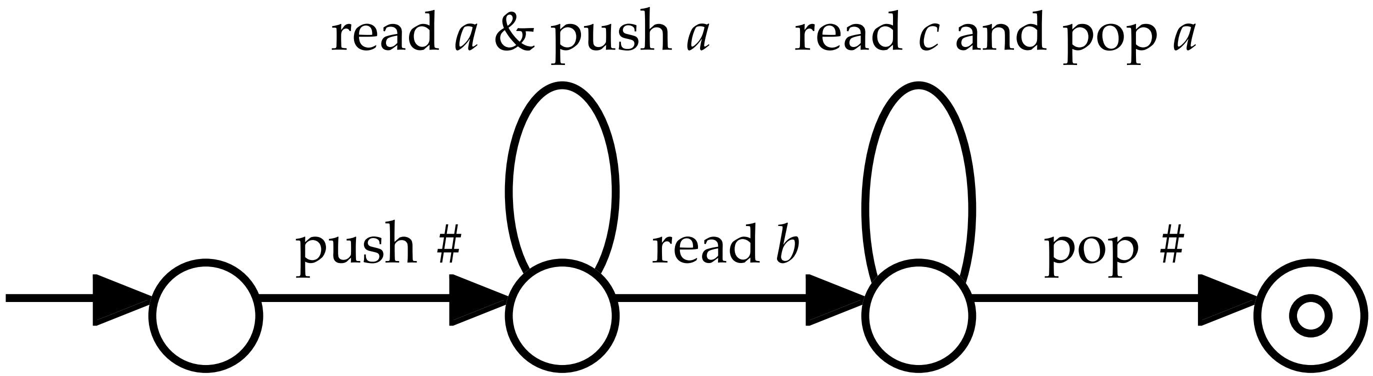
Again, there is a designated *start state* and a set of *accept states*.

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*.


Input
a a a b c c c
Machine M
State q
a
a
#
Stack

For example, the pushdown automaton shown below accepts the language $\{a^n b c^n : n \geq 0\}$.



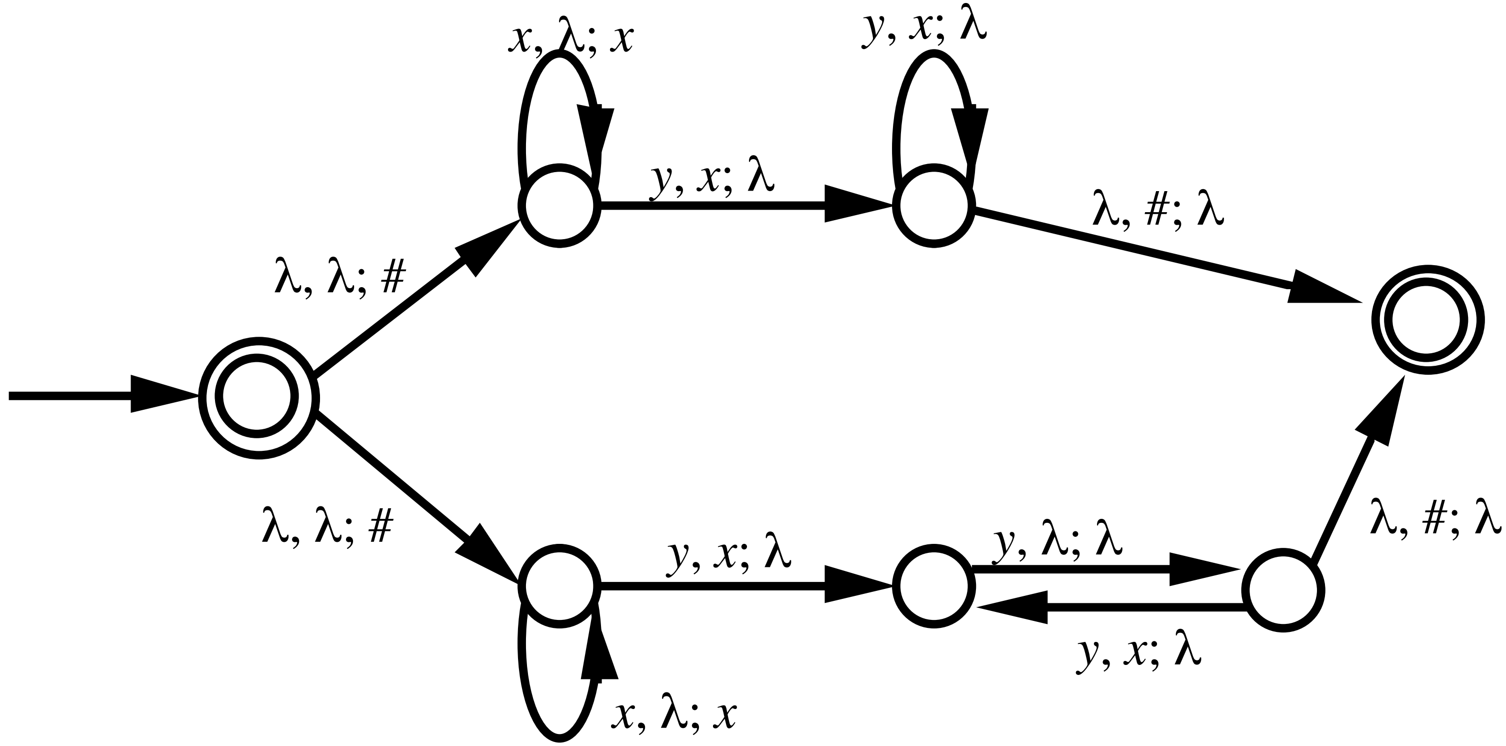
We often label the edges by x, y; z (read x, pop y; push z).

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

$$L = \{x^n y^n : n \geq 0\} \cup \{x^n y^{2n} : n \geq 0\}.$$



L is not accepted by any deterministic pushdown automaton.

Groups

If G is a group and Σ is a (finite) set of symbols together with a surjective (monoid) homomorphism $\varphi : \Sigma^* \to G$, then we say that Σ is a *generating set* for G (we often suppress the reference to φ).

Note that, with this definition, Σ is a *monoid generating set* for G as opposed to a *group generating set*.

The *word problem* of the group G is the set of all the words in Σ^* that represent the identity element of G.

Note that two words α and β in Σ^* represent the same element of G if and only if $\alpha\beta^{-1}$ represents the identity element of G.

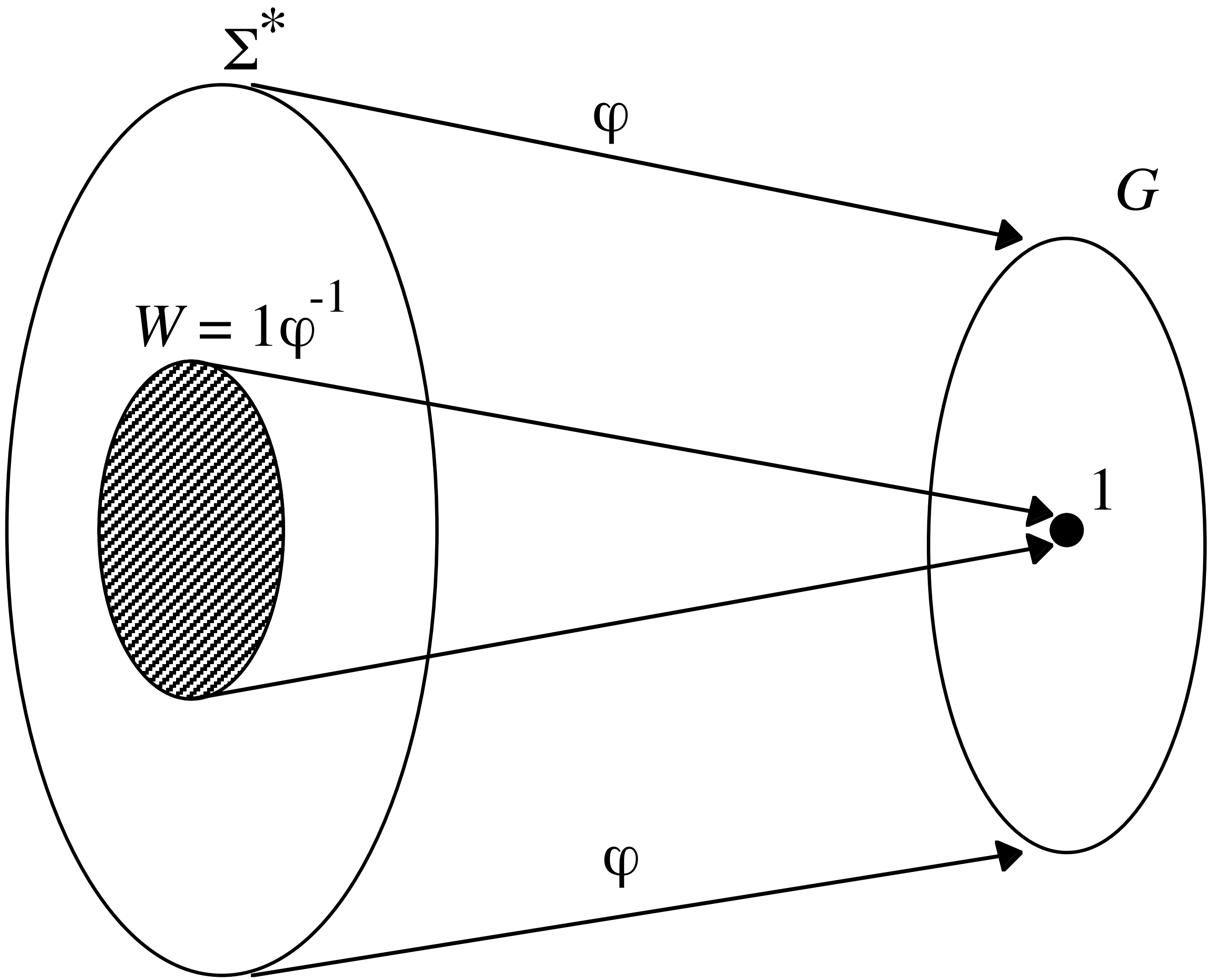
We can form new languages via (monoid) homomorphisms.

If $\varphi : \Sigma^* \to \Omega^*$ is a homomorphism and if $K \subseteq \Sigma^*$ then $K\varphi \subseteq \Omega^*$.

Similarly, if $L \subseteq \Omega^*$ then $L\varphi^{-1} \subseteq \Sigma^*$.

We say that a class of languages $\mathcal{F}$ is *closed under homomorphism* if $K \in \mathcal{F} \Rightarrow K\varphi \in \mathcal{F}$ and that $\mathcal{F}$ is *closed under inverse homomorphism* if $L \in \mathcal{F} \Rightarrow L\varphi^{-1} \in \mathcal{F}$.

If $\mathcal{F}$ is closed under inverse homomorphism and if the word problem of G lies in $\mathcal{F}$ with respect to some finite generating set then the word problem of G lies in $\mathcal{F}$ for all finite generating sets.


Σ^*
φ
G
$W = 1\varphi^{-1}$
1
φ

Word problems and characterizations

Higman. If G is a finitely generated group then the word problem of G is recursively enumerable if and only if G is a subgroup of a finitely presented group.

Boone & Higman. If G is a finitely generated group then the word problem of G is recursive if and only if $G \leq H \leq K$, where H is simple and K is finitely presented.

Anisimov. If G is a finitely generated group then the word problem of G is regular if and only if G is finite.

Muller & Schupp. If G is a finitely generated group then the word problem of G is context-free if and only if G has a free subgroup of finite index.

If a finitely generated group G has a context-free word problem then it has a deterministic context-free word problem.

One-counter languages

One-counter languages – the languages accepted by a *one-counter automaton*, i.e. a (nondeterministic) pushdown automaton where we have only one stack symbol (apart from a bottom marker).

Cone – a 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 the word problem of G lies in $\mathcal{F}$. Then $\mathcal{G}$ is the set of groups with regular, one-counter or context-free word problems.

Herbst. If G is a finitely generated group then the word problem of G is a one-counter language if and only if G has a cyclic subgroup of finite index.

Outside context-free

Holt, Owens & Thomas. The following are equivalent for a finitely generated group G:

(i) The word problem of G is the intersection of n one-counter languages for some $n \geq 1$.

(ii) The word problem of G is the intersection of n deterministic one-counter languages for some $n \geq 1$.

(iii) G is virtually abelian of free abelian rank at most n.

Conjecture - Brough. A finitely generated group G has a word problem which is the intersection of n context-free languages (for some n) if and only if G has a subgroup H of finite index such that H is a finitely generated subgroup of a direct product of free groups.

If this conjecture is true then any such group would have a co-context-free word problem (i.e. a word problem that is the complement of a context-free language).

Groups with a co-context-free word problem have been studied but there are still many unanswered questions about them.

Characterizing word problems

Parkes & Thomas. A language L over an alphabet Σ is the word problem of a group with generating set Σ if and only if L satisfies the following two conditions:

(W1) for all $\alpha \in \Sigma^*$ there exists $\beta \in \Sigma^*$ such that $\alpha\beta \in L$;

(W2) if $\alpha\delta\beta \in L$ and $\delta \in L$ then $\alpha\beta \in L$.

A language satisfying condition (W1) is said to have the *universal prefix property*.

A language satisfying condition (W2) is said to be *deletion closed*.

Decidability

We will now discuss some recent joint work with Sam Jones about decidability questions focussing on the properties (W1) and (W2).

The following problem is decidable:

Input: a finite automaton $M = (Q, \Sigma, \tau, s, A)$.

Output: "yes" if $\text{prefix}(L(M)) = \Sigma^*$;

"no" otherwise.

The following problem is undecidable:

Input: a one-counter automaton $N = (Q, \Sigma, \Gamma, \tau, s, A)$.

Output: "yes" if $\text{prefix}(L(N)) = \Sigma^*$;

"no" otherwise.

Once we have that a problem is undecidable if the input is a one-counter automaton then the problem is also undecidable if the input is a pushdown automaton; so any undecidable property for one-counter languages remains undecidable when we generalize to context-free languages.

The following problem is decidable:

Input: a finite automaton $M = (Q, \Sigma, \tau, s, A)$.

Output: "yes" if $L(M)$ is deletion closed;

"no" otherwise.

The following problem is undecidable:

Input: a one-counter automaton $N = (Q, \Sigma, \Gamma, \tau, s, A)$.

Output: "yes" if $L(N)$ is deletion closed;

"no" otherwise.

The following problem is decidable:

Input: a finite automaton $M = (Q, \Sigma, \tau, s, A)$.

Output: "yes" if $L(M)$ is the word problem of a group;

"no" otherwise.

The following problem is undecidable:

Input: a one-counter automaton $N = (Q, \Sigma, \Gamma, \tau, s, A)$.

Output: "yes" if $L(N)$ the word problem of a group;

"no" otherwise.

Deterministic context-free

If a finitely generated group G has a context-free word problem then it has a deterministic context-free word problem.

A language L over an alphabet Σ is the word problem of a group with generating set Σ if and only if L satisfies the following two conditions:

(W1) for all $\alpha \in \Sigma^*$ there exists $\beta \in \Sigma^*$ such that $\alpha\beta \in L$;

(W2) if $\alpha\delta\beta \in L$ and $\delta \in L$ then $\alpha\beta \in L$.

So we have the following immediate consequence:

> If a context-free language L over an alphabet Σ satisfies the following two conditions:
>
> (W1) for all $\alpha \in \Sigma^*$ there exists $\beta \in \Sigma^*$ such that $\alpha\beta \in L$;
>
> (W2) if $\alpha\delta\beta \in L$ and $\delta \in L$ then $\alpha\beta \in L$,
>
> then L is deterministic context-free.

The following problem is decidable:

Input: a deterministic pushdown automaton $N = (Q, \Sigma, \Gamma, \tau, s, A)$.

Output: "yes" if $L(N)$ the word problem of a group;

"no" otherwise.

Thank You !