The word problem for groups and formal language theory

Fabricio Dos Santos

McGill University

April 2023

Introduction

Group presentation: $G = \langle\, S \mid R \,\rangle$

- S is a set of generators
- R is a set of relations between generators
- Word on S is a concatenation of generators and their inverses

Word Problem: What words on S represent the identity in G?

Conjugacy Problem: What pairs of words on S are conjugate in G?

Table of Contents

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

1. Choose disjoint set of symbols S^{-1} and let $S^{\pm 1} = S \cup S^{-1}$

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

1. Choose disjoint set of symbols S^{-1} and let $S^{\pm 1} = S \cup S^{-1}$
2. Formal inverse: $s \mapsto s^{-1}$ and $s^{-1} \mapsto s$

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

1. Choose disjoint set of symbols S^{-1} and let $S^{\pm 1} = S \cup S^{-1}$
2. Formal inverse: $s \mapsto s^{-1}$ and $s^{-1} \mapsto s$
3. Word on S: $w = s_1 ... s_n$ where $s_i \in S^{\pm 1}$

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

1. Choose disjoint set of symbols S^{-1} and let $S^{\pm 1} = S \cup S^{-1}$
2. Formal inverse: $s \mapsto s^{-1}$ and $s^{-1} \mapsto s$
3. Word on S: $w = s_1 ... s_n$ where $s_i \in S^{\pm 1}$
4. Reduced word: no subword of the form $s^{-1}s$ or ss^{-1}

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

1. Choose disjoint set of symbols S^{-1} and let $S^{\pm 1} = S \cup S^{-1}$
2. Formal inverse: $s \mapsto s^{-1}$ and $s^{-1} \mapsto s$
3. Word on S: $w = s_1 ... s_n$ where $s_i \in S^{\pm 1}$
4. Reduced word: no subword of the form $s^{-1}s$ or ss^{-1}
5. Operation: concatenation + reduction, $u \cdot v = \overline{uv}$

Free groups

Free group: Group generated by a set of elements without any relations between them.

Constructing free group $F(S)$ with basis S:

1. Choose disjoint set of symbols S^{-1} and let $S^{\pm 1} = S \cup S^{-1}$
2. Formal inverse: $s \mapsto s^{-1}$ and $s^{-1} \mapsto s$
3. Word on S: $w = s_1 ... s_n$ where $s_i \in S^{\pm 1}$
4. Reduced word: no subword of the form $s^{-1}s$ or ss^{-1}
5. Operation: concatenation + reduction, $u \cdot v = \overline{uv}$
6. $F(S)$: set of reduced words on S with $\cdot$ operation

Free groups

Definition

A group F is *free* there is some set S such that $F \cong F(S)$.

- S is a set of *free generators*
- $|S|$ is the *rank* of F

Theorem (Schreier)

Subgroups of free groups are free.

Presentations

Definition

Given a group G, the *normal closure* of $A \subseteq G$, denoted $N(A)$, is the smallest normal subgroup of G containing A.

In particular, $N(A)$ is generated by:

$$N(A) = \left\langle \{gag^{-1} : a \in A, g \in G\} \right\rangle$$

Definition

Given a set S and $R \subseteq F(S)$, a group G has *presentation* $\langle\, S \mid R \,\rangle$ if it is isomorphic to the quotient $F(S)/N(R)$.

Presentations (examples)

1. Free group $F(S)$ has presentation $P = \langle\, S \mid \emptyset \,\rangle$.

Presentations (examples)

1. Free group $F(S)$ has presentation $P = \langle\, S \,|\, \emptyset \,\rangle$.
2. Finite cyclic group C_n of order n has presentation $P = \langle\, x \,|\, x^n \,\rangle$.

Presentations (examples)

1. Free group $F(S)$ has presentation $P = \langle\, S \mid \emptyset \,\rangle$.
2. Finite cyclic group C_n of order n has presentation $P = \langle\, x \mid x^n \,\rangle$.
3. Free abelian group of rank 2, $\mathbb{Z}^2$, has presentation
   $P = \left\langle\, x, y \mid xyx^{-1}y^{-1} \,\right\rangle = \langle\, x, y \mid xy = yx \,\rangle$.

Presentations (examples)

1. Free group $F(S)$ has presentation $P = \langle\, S \,|\, \emptyset \,\rangle$.
2. Finite cyclic group C_n of order n has presentation $P = \langle\, x \,|\, x^n \,\rangle$.
3. Free abelian group of rank 2, $\mathbb{Z}^2$, has presentation $P = \left\langle\, x, y \,\middle|\, xyx^{-1}y^{-1} \,\right\rangle = \langle\, x, y \,|\, xy = yx \,\rangle$.
4. Any finite group $G = \{1, \ldots, g_n\}$ has a multiplication table presentation $P = \langle\, 1, \ldots, g_n \,|\, \ldots, g_i g_j = g_k, \ldots \,\rangle$ where we add $g_i g_j = g_k$ for each $1 \leq i, j \leq n$.

Formal language theory

Formal language theory: Study of sets of strings and their expression, especially through automata.

- A set of symbols Σ is called an *alphabet*
- A *word* over Σ is $w = s_1 \ldots s_n$ where $s_i \in \Sigma$
- Σ^* is the set of all words over Σ, and includes empty word ϵ
- A language L is a subset $L \subseteq \Sigma^*$

Hierarchy of languages:

$$\text{Regular} \subsetneq \text{One Counter} \subsetneq \text{Context-free} \subsetneq \text{Recursively Enumerable}$$

The word problem for groups

Decision problem posed by Max Dehn in 1911 [2]:

> Let G be a finitely generated group given by the presentation $\langle S \mid R \rangle$.
> Can we decide whether a given word on S represents the identity in G?

From a language theoretic perspective:

Definition

The *word problem* $\mathrm{WP}(G, \Sigma)$ of a finitely generated group G with presentation $\langle S \mid R \rangle$ is defined as the following set:

$$\mathrm{WP}(G, \Sigma) = \{w \in \Sigma^* : w =_G 1\}$$

where $\Sigma = S^{\pm 1}$.

Regular languages

Definition

For fixed alphabet Σ, a *deterministic finite automaton* (DFA) is a 4-tuple $M = (S, s_0, \delta, F)$ where:

- S is a set of states
- $s_0 \in S$ is a start state
- $\delta : S \times \Sigma \to S$ is a transition function
- $F \subseteq S$ is a set of accept states

A DFA M defines a language $L(M)$ of accepted words in Σ^*.

Definition

A language $L \subseteq \Sigma^*$ is *regular* if it is the language of some DFA.

Regular languages (example)

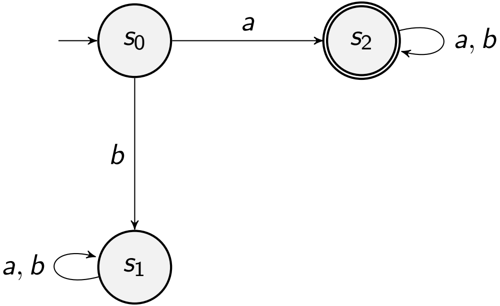


Figure: DFA to recognize words starting in a

Regular word problems

Theorem (Anisimov, 1971 [1])

A group has regular word problem if and only if it is finite.

Proof. Let $G = \{1, ..., g_n\}$. Use multiplication table presentation to create DFA where:

- $\Sigma = \{1, ..., g_n\}$.
- Set of states are group elements $\{1, ..., g_n\}$.
- Start state corresponds to the identity of G.
- Accept state is only the identity.
- In state g_i reading g_j, transition to state g_k where $g_i g_j = g_k$.

M accepts $w = g_{i_1} ... g_{i_m}$ if and only if the product $g_{i_1} ... g_{i_m} = 1$.

Regular word problems

- Now, assume $G = \langle\, S \mid R \,\rangle$ is infinite with regular word problem. There is a DFA M with alphabet $\Sigma = S^{\pm 1}$ such that $L(M) = \mathrm{WP}(G)$.

Regular word problems

- Now, assume $G = \langle\, S \mid R \,\rangle$ is infinite with regular word problem. There is a DFA M with alphabet $\Sigma = S^{\pm 1}$ such that $L(M) = \mathrm{WP}(G)$.

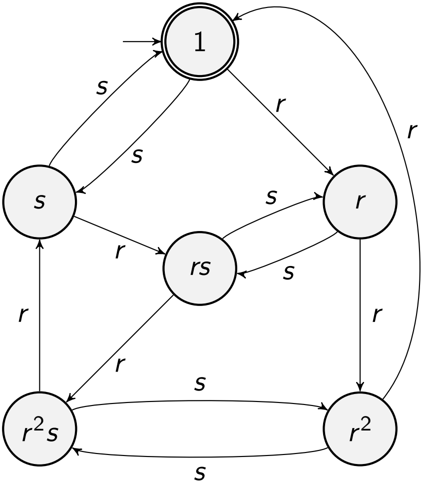
- There are words over Σ of arbitrarily long length such that no nonempty subword represents the identity of G (otherwise G is finite).

Regular word problems

- Now, assume $G = \langle S \mid R \rangle$ is infinite with regular word problem. There is a DFA M with alphabet $\Sigma = S^{\pm 1}$ such that $L(M) = \mathrm{WP}(G)$.
- There are words over Σ of arbitrarily long length such that no nonempty subword represents the identity of G (otherwise G is finite).
- If N is number of states in DFA M, pick some w as above with $|w| > N$. Then, M is in same state after reading two distinct initial segments u, uv of w.

Regular word problems

- Now, assume $G = \langle S \mid R \rangle$ is infinite with regular word problem. There is a DFA M with alphabet $\Sigma = S^{\pm 1}$ such that $L(M) = \mathrm{WP}(G)$.
- There are words over Σ of arbitrarily long length such that no nonempty subword represents the identity of G (otherwise G is finite).
- If N is number of states in DFA M, pick some w as above with $|w| > N$. Then, M is in same state after reading two distinct initial segments u, uv of w.
- However, uu^{-1} is accepted by M but uvu^{-1} is not $\implies$ contradiction.



$$D_3 = \langle\, r, s \mid s^2, r^3, (rs)^2 \,\rangle$$

$$\Sigma = \{r, s, r^{-1}, s^{-1}\}$$

Context-free languages

Regular languages are limited: $L = \{a^n b^n : n \in \mathbb{N}\}$ is not regular.

Nondeterministic Pushdown automaton (PDA):

- DFA + Stack + Nondeterminism
- Stack:
  - Stack alphabet Z
  - Transition involves reading symbol from top of stack, deleting it, and pushing some word over Z
- Nondeterminism:
  - Different possible moves for same input letter
  - ϵ-moves

Context-free languages

M accepts a word $w \in \Sigma^*$ if after reading w it is possible that M is in an accept state with empty stack.

A PDA M defines a language $L(M)$ of accepted words.

Definition

A language $L \subseteq \Sigma^*$ is *context-free* if it is the language of some PDA.

Unlike the case of regular languages, deterministic and nondeterministic PDA lead to different classes of languages.

Context-free languages (example)

If transition is not written reject word.

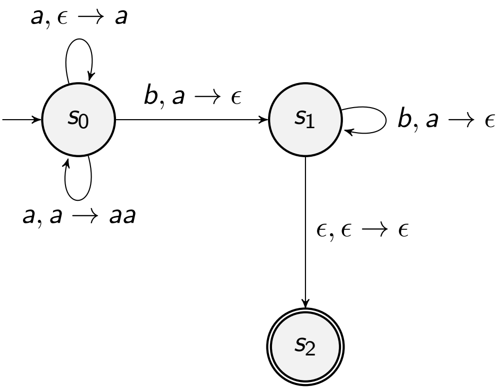


Figure: PDA to recognize $L = \{a^n b^n : n \in \mathbb{N}\}$

Context-free word problems

Since regular languages are subset of context-free languages, finite groups have context-free word problem. However, these are not the only ones.

Definition

Given a property P, a group G is *virtually* P if it has a subgroup of finite index in G which has the property P.

We will prove that virtually free groups have context-free word problem. In particular, this is the case for free groups.

Example: F_n

- Free group of rank n has presentation $F_n = \langle\, x_1, \ldots, x_n \,|\, \emptyset \,\rangle$.

Example: F_n

- Free group of rank n has presentation $F_n = \langle\, x_1, \ldots, x_n \,|\, \emptyset \,\rangle$.
- Word w on $X = \{x_1, \ldots, x_n\}$ represents the identity $\iff$ we can successively delete subwords of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$ until w becomes empty.

Example: F_n

- Free group of rank n has presentation $F_n = \langle\, x_1, ..., x_n \,|\, \emptyset \,\rangle$.
- Word w on $X = \{x_1, ..., x_n\}$ represents the identity $\iff$ we can successively delete subwords of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$ until w becomes empty.
- Create PDA with only one state and which uses stack as follows: when reading input letter x_i^{ε} ($\varepsilon \in \{\pm 1\}$), it pops stack when its inverse is on top, or otherwise it pushes x_i^{ε} on top of the stack.

Two important lemmas for next theorem and future theorems.

Lemma 1

Let G be a group and H a subgroup of G of finite index. Then, there exists a normal subgroup $N \subseteq H$ in G, which is also of finite index in G.

Lemma 2

Let G be a finitely generated group, and H a subgroup of G of finite index. Then, H is also finitely generated.

Context-free word problems

Theorem (Muller and Schupp, 1983 [6])

A finitely generated group G has context-free word problem if and only G is virtually free.

Proof.

- G is virtually free, so it has a free subgroup of finite index in G.

Theorem (Muller and Schupp, 1983 [6])

A finitely generated group G has context-free word problem if and only G is virtually free.

Proof.

- G is virtually free, so it has a free subgroup of finite index in G.
- Use two previous lemmas to extract a finitely generated free subgroup N, which is normal and has finite index in G.

Theorem (Muller and Schupp, 1983 [6])

A finitely generated group G has context-free word problem if and only G is virtually free.

Proof.

- G is virtually free, so it has a free subgroup of finite index in G.
- Use two previous lemmas to extract a finitely generated free subgroup N, which is normal and has finite index in G.
- Let $y_1, \ldots, y_n$ be free generators of N, and $d_1, \ldots, d_t$ be coset representatives for each right coset of N.

Context-free word problems

Theorem (Muller and Schupp, 1983 [6])

A finitely generated group G has context-free word problem if and only G is virtually free.

Proof.

- G is virtually free, so it has a free subgroup of finite index in G.
- Use two previous lemmas to extract a finitely generated free subgroup N, which is normal and has finite index in G.
- Let $y_1, \dots, y_n$ be free generators of N, and $d_1, \dots, d_t$ be coset representatives for each right coset of N.
- PDA M has input alphabet $\{y_1, \dots, y_n, d_1, \dots, d_t\}^{\pm 1}$. Keeps track of the y's on stack and d's using states.

Context-free word problems

- Since N is normal in G, for each y_j and d_i we get that $d_i y_j d_i^{-1} = u_{i,j}$ for some $u_{i,j} \in N$.

Context-free word problems

- Since N is normal in G, for each y_j and d_i we get that $d_i y_j d_i^{-1} = u_{i,j}$ for some $u_{i,j} \in N$.

- By considering right coset multiplication, for each pair d_i, d_j and $\varepsilon \in \{\pm 1\}$, we get that: $d_i d_j^{\varepsilon} = z_{i,\varepsilon,j} d_k$ for some $z_{i,\varepsilon,j} \in N$.

Context-free word problems

- Since N is normal in G, for each y_j and d_i we get that $d_i y_j d_i^{-1} = u_{i,j}$ for some $u_{i,j} \in N$.
- By considering right coset multiplication, for each pair d_i, d_j and $\varepsilon \in \{\pm 1\}$, we get that: $d_i d_j^{\varepsilon} = z_{i,\varepsilon,j} d_k$ for some $z_{i,\varepsilon,j} \in N$.
- Transform any word into the form wd_i where w is a freely reduced word in the y's.

Context-free word problems

- The states of M are $\{d_1, ..., d_t\}$, where start and accept state is d_1 (identity).

Context-free word problems

- The states of M are $\{d_1, ..., d_t\}$, where start and accept state is d_1 (identity).
- Uses two relations to transition states, and process words onto stack as in example of free groups:

Context-free word problems

- The states of M are $\{d_1, ..., d_t\}$, where start and accept state is d_1 (identity).
- Uses two relations to transition states, and process words onto stack as in example of free groups:
  - When reading y_j^ϵ in state d_i, process w onto stack, where $d_i y_j^\epsilon = w d_i$.

Context-free word problems

- The states of M are $\{d_1, ..., d_t\}$, where start and accept state is d_1 (identity).
- Uses two relations to transition states, and process words onto stack as in example of free groups:
  - When reading y_j^ϵ in state d_i, process w onto stack, where $d_i y_j^\epsilon = w d_i$.
  - When reading d_j^ϵ in state d_i, move to state d_k and process w onto stack, where $d_i d_j^\epsilon = w d_k$.

Context-free word problems

- The states of M are $\{d_1, ..., d_t\}$, where start and accept state is d_1 (identity).
- Uses two relations to transition states, and process words onto stack as in example of free groups:
  - When reading y_j^ϵ in state d_i, process w onto stack, where $d_i y_j^\epsilon = w d_i$.
  - When reading d_j^ϵ in state d_i, move to state d_k and process w onto stack, where $d_i d_j^\epsilon = w d_k$.
- A word wd_i is accepted $\iff$ w is empty and $d_i = d_1$.

One counter word problems

If we restrict the stack alphabet of a PDA to only 1 stack symbol, we get *one counter languages*. In particular, we have:

$$\text{Regular} \subsetneq \text{One counter} \subsetneq \text{Context-free}$$

- $\{a^n b^n : n \in \mathbb{N}\}$ is one counter.
- $\{a^n b^m a^m b^n : n \in \mathbb{N}\}$ is context-free but not one counter.

One counter word problems

Theorem (Herbst, 1991 [3])

A finitely generated group G has one counter word problem if and only G is virtually cyclic.

Proof.

Theorem (Herbst, 1991 [3])

A finitely generated group G has one counter word problem if and only G is virtually cyclic.

Proof.

- Using lemmas 1 and 2, we can extract a normal subgroup $Z = \langle x \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.

One counter word problems

Theorem (Herbst, 1991 [3])

A finitely generated group G has one counter word problem if and only G is virtually cyclic.

Proof.

- Using lemmas 1 and 2, we can extract a normal subgroup $Z = \langle x \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.
- Construction of the PDA M is similar, but has only one stack symbol x. But since x^{-1} is not a stack symbol, there are problems when trying to process it onto an empty stack.

One counter word problems

Theorem (Herbst, 1991 [3])

A finitely generated group G has one counter word problem if and only G is virtually cyclic.

Proof.

- Using lemmas 1 and 2, we can extract a normal subgroup $Z = \langle x \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.
- Construction of the PDA M is similar, but has only one stack symbol x. But since x^{-1} is not a stack symbol, there are problems when trying to process it onto an empty stack.
- Consider instead two copies of each state of the PDA M, one "positive" and one "negative". These states indicate if stack symbol is interpreted as x or x^{-1}.

Recursively enumerable languages

$\mathbb{Z}^2 = \langle\, x, y \mid xyx^{-1}y^{-1} \,\rangle$ is the "simplest" group which is not context-free.

We need a more general class of languages and more powerful automata to recognize these word problems.

Recursively enumerable language: language L such that there exists an algorithm that lists all the members of L, and only the members of L, in some arbitrary order.

Recursively enumerable languages

The notion of algorithm is formalized by a *Turing machine* (TM).

A TM may accept, reject or never halt when reading an input word.

Definition

A language L is *recursively enumerable* (RE) if it is the set of accepted words of some TM.

Turing machines are complicated to work with, so it is often enough to describe an enumerating procedure for the language.

Recursively enumerable word problems

Definition

A finitely generated group is *recursively presentable* if it has a presentation of the type $\langle s_1, \ldots, s_n \mid R \rangle$ where R is a recursively enumerable set of words on the generators.

Recursively presentable groups are characterized by Higman's embedding theorem.

Theorem (Higman, 1961 [4])

A finitely generated group G is recursively presentable if and only if it embeds into some finitely presented group.

Recursively enumerable word problems

Theorem

The word problem $\mathrm{WP}(G)$ of a finitely generated group G is recursively enumerable if and only if G embeds into some finitely presented group.

Proof.

Recursively enumerable word problems

Theorem

The word problem $\mathrm{WP}(G)$ of a finitely generated group G is recursively enumerable if and only if G embeds into some finitely presented group.

Proof.

- Let $G = \langle X \rangle$ and $\mathrm{WP}(G)$ be RE.

Recursively enumerable word problems

Theorem

The word problem $\mathrm{WP}(G)$ of a finitely generated group G is recursively enumerable if and only if G embeds into some finitely presented group.

Proof.

- Let $G = \langle X \rangle$ and $\mathrm{WP}(G)$ be RE.
- We can easily show that $G \cong G' = \langle\, X \mid \mathrm{WP}(G)\, \rangle$.

Recursively enumerable word problems

Theorem

The word problem $\mathrm{WP}(G)$ of a finitely generated group G is recursively enumerable if and only if G embeds into some finitely presented group.

Proof.

- Let $G = \langle X \rangle$ and $\mathrm{WP}(G)$ be RE.
- We can easily show that $G \cong G' = \langle\, X \mid \mathrm{WP}(G) \,\rangle$.
- $\langle\, X \mid \mathrm{WP}(G) \,\rangle$ is a recursive presentation, so G embeds into some finitely presented group.

Recursively enumerable word problems

- Assume G has recursive presentation $\langle S \mid R \rangle$. Note that $G \cong F(S)/\mathrm{WP}(G)$ since $N(R) = \mathrm{WP}(G)$ by definition.

Recursively enumerable word problems

- Assume G has recursive presentation $\langle\, S \mid R \,\rangle$. Note that $G \cong F(S)/\mathrm{WP}(G)$ since $N(R) = \mathrm{WP}(G)$ by definition.

- Therefore, $w \in \mathrm{WP}(G) \iff w$ is the concatenation of some $r_i \in R$ conjugated by some $w_i \in F(S)$.

Recursively enumerable word problems

- Assume G has recursive presentation $\langle S \mid R \rangle$. Note that $G \cong F(S)/\mathrm{WP}(G)$ since $N(R) = \mathrm{WP}(G)$ by definition.
- Therefore, $w \in \mathrm{WP}(G) \iff w$ is the concatenation of some $r_i \in R$ conjugated by some $w_i \in F(S)$.
- Given enumerations $r_1, r_2, \ldots$ for R and $w_1, w_2, \ldots$ for $F(S)$, we can enumerate $\mathrm{WP}(G)$ as follows.

Recursively enumerable word problems

- Assume G has recursive presentation $\langle S \mid R \rangle$. Note that $G \cong F(S)/\mathrm{WP}(G)$ since $N(R) = \mathrm{WP}(G)$ by definition.
- Therefore, $w \in \mathrm{WP}(G) \iff w$ is the concatenation of some $r_i \in R$ conjugated by some $w_i \in F(S)$.
- Given enumerations $r_1, r_2, \ldots$ for R and $w_1, w_2, \ldots$ for $F(S)$, we can enumerate $\mathrm{WP}(G)$ as follows.
- For $t = 1, 2, \ldots$ produce all words with $n \leq t$ using $r_1, \ldots, r_t$ and $w_1, \ldots, w_t$.

Summary

Theorem

Given a finitely generated group G, $\mathrm{WP}(G)$ is:

$$\begin{aligned}
\text{Regular} &\iff G \text{ is Finite} \\
\text{One counter} &\iff G \text{ is virtually cyclic} \\
\text{Context-free} &\iff G \text{ is virtually free} \\
\text{Recursively enumerable} &\iff G \text{ is recursively presentable}
\end{aligned}$$

The conjugacy problem for groups

Another decision problem posed by Max Dehn in 1911 [2]:

> Let G be a finitely generated group given by the presentation $\langle S \mid R \rangle$. Can we decide whether two words in S are conjugate in G?

From a language theoretic perspective: we are interested in pairs (u, v) of words on S such that u is conjugate to v.

If v is restricted to the empty word, we get the word problem of G. Thus, the conjugacy problem should be at least as hard as the word problem.

Two-variable machines

We consider two variable machines that read pairs (u, v) of words over X, where shortest word is padded at the end by \$.

Synchronous case is equivalent to one variable machine with alphabet $\Sigma = (X \cup \{\$\}) \times (X \cup \{\$\})$.

Definition

The *conjugacy problem* $\mathrm{CP}(G, \Sigma)$ of a finitely generated group G with presentation $\langle\, S \mid R \,\rangle$ is defined as the following set:

$$\mathrm{CP}(G, \Sigma) = \{w = (x_1, y_1)...(x_n, y_n) \in \Sigma^* : \exists\, g \in G, gx_1...x_n g^{-1} =_G y_1...y_n\}$$

where $\Sigma = (S^{\pm 1} \cup \{\$\})^2$.

Synchronously regular conjugacy problem

Theorem (Holt, Rees, Röver [5])

A group G has synchronously regular conjugacy problem if and only if it is finite.

Proof.

- If G has synchronously regular conjugacy problem, we can modify the DFA for the conjugacy problem to accept the word problem of G.

Synchronously regular conjugacy problem

Theorem (Holt, Rees, Röver [5])

A group G has synchronously regular conjugacy problem if and only if it is finite.

Proof.

- If G has synchronously regular conjugacy problem, we can modify the DFA for the conjugacy problem to accept the word problem of G.
- If G is finite, we can create a DFA based on the Cayley graph of $G \times G$ with accept states (u, v) where u is conjugate to v.

Synchronously one counter conjugacy problem

Theorem (Holt, Rees, Röver [5])

Every virtually cyclic group has synchronously one counter conjugacy problem.

Proof.

Synchronously one counter conjugacy problem

Theorem (Holt, Rees, Röver [5])

Every virtually cyclic group has synchronously one counter conjugacy problem.

Proof.

- We can again extract a normal subgroup $Z = \langle z \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.

Theorem (Holt, Rees, Röver [5])

Every virtually cyclic group has synchronously one counter conjugacy problem.

Proof.

- We can again extract a normal subgroup $Z = \langle z \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.
- PDA has input alphabet $(T \cup \{z\})^{\pm 1}$ where T is a set of coset representatives.

Theorem (Holt, Rees, Röver [5])

Every virtually cyclic group has synchronously one counter conjugacy problem.

Proof.

- We can again extract a normal subgroup $Z = \langle z \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.
- PDA has input alphabet $(T \cup \{z\})^{\pm 1}$ where T is a set of coset representatives.
- Each word w can be written in normal form $z^k t$ for some $k \in \mathbb{Z}, t \in T$.

Theorem (Holt, Rees, Röver [5])

Every virtually cyclic group has synchronously one counter conjugacy problem.

Proof.

- We can again extract a normal subgroup $Z = \langle z \rangle$ of finite index in G which is isomorphic to $\mathbb{Z}$.
- PDA has input alphabet $(T \cup \{z\})^{\pm 1}$ where T is a set of coset representatives.
- Each word w can be written in normal form $z^k t$ for some $k \in \mathbb{Z}, t \in T$.
- On input $(z^i r, z^j s)$, the PDA guesses some $z^l t \in T$ and checks that:
  1. r^t and s are in the same coset
  2. exponents match depending on different cases

Summary

Theorem

Given a finitely generated group G, $\mathrm{CP}(G)$ is:

$$\begin{aligned} \text{Synchronously regular} &\iff G \text{ is Finite} \\ \text{Synchronously one counter} &\iff G \text{ is virtually cyclic} \end{aligned}$$

References I

A. Anisimov.
Groups languages.
Kybernetika, 4:18–24, 1971.

M. Dehn.
Über unendliche diskontinuierliche gruppen.
Mathematische Annalen, 71:116–144, 1912.

T. Herbst.
On a subclass of context-free groups.
RAIRO Theor. Informatics Appl., 25:255–272, 1991.

G. Higman.
Subgroups of finitely presented groups.
Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 262:455 – 475, 1961.

References II

D. Holt, S. Rees, and C. Röver.
Groups with context-free conjugacy problems.
IJAC, 21:193–216, 02 2011.

D. E. Muller and P. E. Schupp.
Groups, the theory of ends, and context-free languages.
Journal of Computer and System Sciences, 26(3):295–310, 1983.