The freeness problem on matrix groups

Summer research 2021 (Supervisor: Emmanuel Breuillard)

Georgi Kocharyan

University of Cambridge

October 12, 2021

Freeness problem

We are given a semigroup S.

Freeness problem

We are given a semigroup S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Freeness problem

We are given a semigroup S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve Free$[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

Turing machines

Important preliminary: Concepts of *algorithms* and *undecidability*.

Turing machines

Important preliminary: Concepts of *algorithms* and *undecidability*.
Turing 1936: Turing machine (TM) as model of computation, consisting of:

Turing machines

Important preliminary: Concepts of *algorithms* and *undecidability*.
Turing 1936: Turing machine (TM) as model of computation, consisting of:

- An (infinite) tape of 0s and 1s, and a tape head

Turing machines

Important preliminary: Concepts of *algorithms* and *undecidability*.
Turing 1936: Turing machine (TM) as model of computation, consisting of:

- An (infinite) tape of 0s and 1s, and a tape head
- A finite amount of 'states' dictating the behaviour of the TM depending on if head is on a 0 or a 1

Turing machines

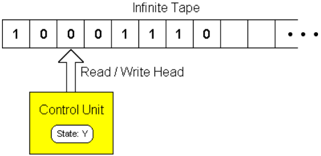


Figure: Depiction of a TM [1]

[1] http://science.slc.edu/ jmarshall/courses/2002/fall/cs30/Lectures/week08/TuringMachine.gif

Turing machines

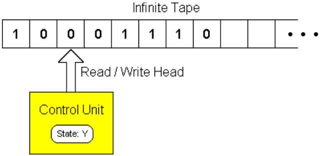


Figure: Depiction of a TM [1]

- Can change character under tape head (i.e. 0 to 1)

[1] http://science.slc.edu/ jmarshall/courses/2002/fall/cs30/Lectures/week08/TuringMachine.gif

Turing machines

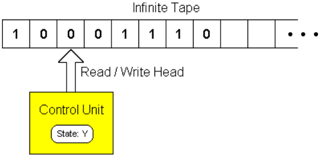


Figure: Depiction of a TM [1]

- Can change character under tape head (i.e. 0 to 1)
- Can move tape head right or left

[1] http://science.slc.edu/ jmarshall/courses/2002/fall/cs30/Lectures/week08/TuringMachine.gif

Turing machines

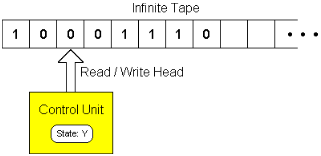


Figure: Depiction of a TM [1]

- Can change character under tape head (i.e. 0 to 1)
- Can move tape head right or left
- Can change state

[1] http://science.slc.edu/ jmarshall/courses/2002/fall/cs30/Lectures/week08/TuringMachine.gif

Turing machines

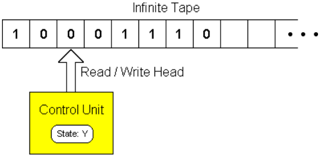


Figure: Depiction of a TM [1]

- Can change character under tape head (i.e. 0 to 1)
- Can move tape head right or left
- Can change state

A problem is *undecidable* if there exists no Turing machine always outputting the correct solution to it.

[1] http://science.slc.edu/ jmarshall/courses/2002/fall/cs30/Lectures/week08/TuringMachine.gif

Halting problem

First example of an undecidable problem.

Halting problem

First example of an undecidable problem.
Halting problem: Can we find a TM that can, given another TM, always predict whether or not this TM will halt on a given input?

Halting problem

First example of an undecidable problem.

Halting problem: Can we find a TM that can, given another TM, always predict whether or not this TM will halt on a given input?

Theorem (Turing, 1938)

The halting problem is undecidable.

Halting problem

First example of an undecidable problem.

Halting problem: Can we find a TM that can, given another TM, always predict whether or not this TM will halt on a given input?

Theorem (Turing, 1938)

The halting problem is undecidable.

Proof.

There are countably many Turing machines, call them T_n.

Halting problem

First example of an undecidable problem.

Halting problem: Can we find a TM that can, given another TM, always predict whether or not this TM will halt on a given input?

Theorem (Turing, 1938)

The halting problem is undecidable.

Proof.

There are countably many Turing machines, call them T_n. Let S be the set of natural numbers n for which T_n halts on input n. If there existed a TM T solving the halting problem, we could easily get a TM T' which halts if on input k if and only if $k \notin S$.

Halting problem

First example of an undecidable problem.

Halting problem: Can we find a TM that can, given another TM, always predict whether or not this TM will halt on a given input?

Theorem (Turing, 1938)

The halting problem is undecidable.

Proof.

There are countably many Turing machines, call them T_n. Let S be the set of natural numbers n for which T_n halts on input n. If there existed a TM T solving the halting problem, we could easily get a TM T' which halts if on input k if and only if $k \notin S$. Now since $T' = T_m$, we see that T_m halting on m implies it doesn't and vice versa. $\square$

Post correspondence problem

One of the first natural examples for an undecidable problem is the *Post correspondence problem* (PCP).

Post correspondence problem

One of the first natural examples for an undecidable problem is the *Post correspondence problem* (PCP).
Fix an alphabet $\Sigma = \{a, b\}$.

Post correspondence problem

One of the first natural examples for an undecidable problem is the *Post correspondence problem* (PCP).
Fix an alphabet $\Sigma = \{a, b\}$. We are given a finite set of n dominoes, each bearing a word in Σ on each of its sides.

Post correspondence problem

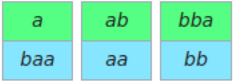
One of the first natural examples for an undecidable problem is the *Post correspondence problem* (PCP).
Fix an alphabet $\Sigma = \{a, b\}$. We are given a finite set of n dominoes, each bearing a word in Σ on each of its sides.
Question. Can we, given an unlimited supply of each domino, line a non-zero amount of them up so that the top and bottom spell out the same word?

Post correspondence problem

Examples:

Post correspondence problem

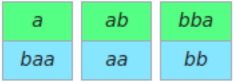
Examples:



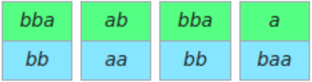
(a) Can we spell the same word on top and bottom?

Post correspondence problem

Examples:



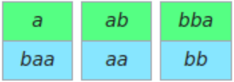
(a) Can we spell the same word on top and bottom?



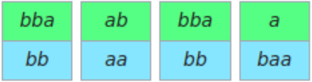
(b) Yes, this works! (The solution is 3,2,3,1)

Post correspondence problem

Examples:



(a) Can we spell the same word on top and bottom?



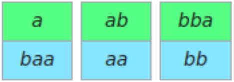
(b) Yes, this works! (The solution is 3,2,3,1)

Theorem (Post, 1946)

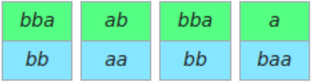
The PCP is undecidable.

Post correspondence problem

Examples:



(a) Can we spell the same word on top and bottom?



(b) Yes, this works! (The solution is 3,2,3,1)

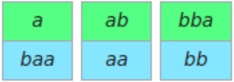
Theorem (Post, 1946)

The PCP is undecidable.

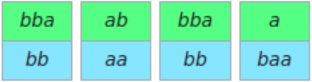
What if we fix n?

Post correspondence problem

Examples:



(a) Can we spell the same word on top and bottom?



(b) Yes, this works! (The solution is 3,2,3,1)

Theorem (Post, 1946)

The PCP is undecidable.

What if we fix n? In this case the PCP becomes decidable for $n \leq 2$, but remains undecidable for $n \geq 5$.

Freeness problem

We are given a (semi)group S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve Free$[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

Freeness problem

We are given a (semi)group S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \dots, x_n, y_1, \dots, y_m \in X$ we have

$$(x_1, \dots, x_n) = (y_1, \dots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve Free$[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

- $\mathbb{W}$ free semigroup: decidable (Sardinas-Patterson algorithm)

Freeness problem

We are given a (semi)group S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve Free$[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

- $\mathbb{W}$ free semigroup: decidable (Sardinas-Patterson algorithm)
- $\mathbb{F}$ free group: decidable (Stallings foldings)

Freeness problem

We are given a (semi)group S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve $\mathrm{Free}[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

- $\mathbb{W}$ free semigroup: decidable (Sardinas-Patterson algorithm)
- $\mathbb{F}$ free group: decidable (Stallings foldings)
- $\mathbb{W} \times \mathbb{W}$: undecidable! (via MMPCP)

Freeness problem

We are given a (semi)group S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve Free$[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

- $\mathbb{W}$ free semigroup: decidable (Sardinas-Patterson algorithm)
- $\mathbb{F}$ free group: decidable (Stallings foldings)
- $\mathbb{W} \times \mathbb{W}$: undecidable! (via MMPCP) (in particular Free[9])

Freeness problem

We are given a (semi)group S.

Definition (Code)

$X \subseteq S$ is called a *code* if for any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$ we have

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \iff x_1 \cdots x_n = y_1 \cdots y_m.$$

Can we solve Free$[k](S)$, i.e. find an algorithm that decides for every possible $k-$subset of S if it is a code?

- $\mathbb{W}$ free semigroup: decidable (Sardinas-Patterson algorithm)
- $\mathbb{F}$ free group: decidable (Stallings foldings)
- $\mathbb{W} \times \mathbb{W}$: undecidable! (via MMPCP) (in particular Free[9])
- $\mathbb{F} \times \mathbb{F}$: decidable (exercise)

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Paterson, 1970: Can embed $\mathbb{W} \times \mathbb{W}$ into $\mathbb{N}^{3\times 3}$

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Paterson, 1970: Can embed $\mathbb{W} \times \mathbb{W}$ into $\mathbb{N}^{3\times 3}$
  The reverse $n-$ary representation of a word in $\mathbb{W}$:

$$\sigma(a_{i_1} \cdots a_{i_k}) = \sum_{j=1}^{k} i_j n^{k-j}$$

$$\gamma : \mathbb{W} \times \mathbb{W} \to \mathbb{N}^{3\times 3}$$

$$(u, v) \mapsto \begin{pmatrix} n^{|u|} & 0 & 0 \\ 0 & n^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix}$$

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Paterson, 1970: Can embed $\mathbb{W} \times \mathbb{W}$ into $\mathbb{N}^{3\times 3}$
  The reverse $n-$ary representation of a word in $\mathbb{W}$:

$$\sigma(a_{i_1} \cdots a_{i_k}) = \sum_{j=1}^{k} i_j n^{k-j}$$

$$\gamma : \mathbb{W} \times \mathbb{W} \to \mathbb{N}^{3\times 3}$$

$$(u, v) \mapsto \begin{pmatrix} n^{|u|} & 0 & 0 \\ 0 & n^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix}$$

  Cassaigne, Harju, Karhumäki (1999): No such morphism exists for $\mathbb{C}^{2\times 2}$.

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Fairly easy (see Lyndon-Ullman problem): Can embed $\mathbb{F} \times \mathbb{F}$ into $\mathrm{GL}_4(\mathbb{Z})$

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Fairly easy (see Lyndon-Ullman problem): Can embed $\mathbb{F} \times \mathbb{F}$ into $\mathrm{GL}_4(\mathbb{Z})$
- Result from placement (2021): No such morphism exists for $\mathbb{C}^{3\times 3}$.

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Fairly easy (see Lyndon-Ullman problem): Can embed $\mathbb{F} \times \mathbb{F}$ into $\mathrm{GL}_4(\mathbb{Z})$
- Result from placement (2021): No such morphism exists for $\mathbb{C}^{3\times 3}$.
- So we see that the freeness problem is undecidable for semigroups generated by 9 $3 \times 3-$matrices!

Why we care about direct products

We can embed certain direct products into matrix (semi)groups, which preserves undecidability of most decision problems.

- Fairly easy (see Lyndon-Ullman problem): Can embed $\mathbb{F} \times \mathbb{F}$ into $\mathrm{GL}_4(\mathbb{Z})$
- Result from placement (2021): No such morphism exists for $\mathbb{C}^{3\times 3}$.
- So we see that the freeness problem is undecidable for semigroups generated by 9 $3 \times 3-$matrices!
- For matrix <u>groups</u>: open problem

Further matrix group problems

Let G be a finitely generated matrix group.

Further matrix group problems

Let G be a finitely generated matrix group.

Definition (Word problem)

Given a word w in the generators of G, decide if $w = 1$ or not.

Further matrix group problems

Let G be a finitely generated matrix group.

Definition (Word problem)

Given a word w in the generators of G, decide if $w = 1$ or not.

Definition (Membership problem or generalised word problem)

Given a finite set of words $\{g_1, \ldots, g_k\}$ in the generators and another word w, determine whether $w \in \langle g_1, \ldots, g_k \rangle$ or not.

Further matrix group problems

Let G be a finitely generated matrix group.

Definition (Word problem)

Given a word w in the generators of G, decide if $w = 1$ or not.

Definition (Membership problem or generalised word problem)

Given a finite set of words $\{g_1, \ldots, g_k\}$ in the generators and another word w, determine whether $w \in \langle g_1, \ldots, g_k \rangle$ or not.

Definition (Conjugacy problem)

Given two words in the generators determines whether they are conjugate *within* G or not.

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Definition

The *Mihailova subgroup* $M(G)$ of $\mathbb{F} \times \mathbb{F}$ based on a given finitely generated group G is the subgroup $\{(x, y) \in \mathbb{F} \times \mathbb{F} |\ x = y \text{ in } G\}$.

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Definition

The *Mihailova subgroup* $M(G)$ of $\mathbb{F} \times \mathbb{F}$ based on a given finitely generated group G is the subgroup $\{(x, y) \in \mathbb{F} \times \mathbb{F} | \ x = y \text{ in } G\}$.

Theorem

The membership problem is undecidable for groups of $4 \times 4-$*matrices.*

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Definition

The *Mihailova subgroup* $M(G)$ of $\mathbb{F} \times \mathbb{F}$ based on a given finitely generated group G is the subgroup $\{(x, y) \in \mathbb{F} \times \mathbb{F} | \, x = y \text{ in } G\}$.

Theorem

The membership problem is undecidable for groups of $4 \times 4-$*matrices.*

Proof.

If G has undecidable word problem, then $w = 1$ in G is equivalent to $(w, 1) \in M(G)$.

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Definition

The *Mihailova subgroup* $M(G)$ of $\mathbb{F} \times \mathbb{F}$ based on a given finitely generated group G is the subgroup $\{(x, y) \in \mathbb{F} \times \mathbb{F} | \; x = y \text{ in } G\}$.

Theorem

The membership problem is undecidable for groups of $4 \times 4-$*matrices.*

Proof.

If G has undecidable word problem, then $w = 1$ in G is equivalent to $(w, 1) \in M(G)$. Then just embed $M(G)$ into $\mathrm{GL}_4(\mathbb{Z})$. $\square$

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Definition

The *Mihailova subgroup* $M(G)$ of $\mathbb{F} \times \mathbb{F}$ based on a given finitely generated group G is the subgroup $\{(x, y) \in \mathbb{F} \times \mathbb{F} |\ x = y \text{ in } G\}$.

Theorem

The conjugacy problem is undecidable for groups of $4 \times 4-$*matrices.*

Further matrix group problems

Can show undecidability of both latter problems using the *Mihailova construction*.

Definition

The *Mihailova subgroup* $M(G)$ of $\mathbb{F} \times \mathbb{F}$ based on a given finitely generated group G is the subgroup $\{(x, y) \in \mathbb{F} \times \mathbb{F} | \ x = y \text{ in } G\}$.

Theorem

The conjugacy problem is undecidable for groups of $4 \times 4-$*matrices.*

Proof.

Add x to the generators and relators of G. Then $w = 1$ in G if and only if (x, x) is conjugate to $(x, w^{-1}xw)$ in $M(G)$. (Exercise!) □

Hyperplane problem

Definition

The *matrix mortality problem* for a matrix semigroup G asks for an algorithm which given a finite set of matrices $\{G_1, \ldots, G_k\} \subseteq G$ determines whether $0 \in \{G_1, \ldots, G_k\}^+$.

Hyperplane problem

Definition

The *matrix mortality problem* for a matrix semigroup G asks for an algorithm which given a finite set of matrices $\{G_1, \ldots, G_k\} \subseteq G$ determines whether $0 \in \{G_1, \ldots, G_k\}^+$.

Definition

The *upper-left-corner problem* or *hyperplane problem* for a matrix semigroup G asks for an algorithm which given a finite set of matrices $\{G_1, \ldots, G_k\} \subseteq G$ determines whether $\exists M \in \{G_1, \ldots, G_k\}^+$ with $M_{11} = 0$.

Hyperplane problem

Definition

The *matrix mortality problem* for a matrix semigroup G asks for an algorithm which given a finite set of matrices $\{G_1, \dots, G_k\} \subseteq G$ determines whether $0 \in \{G_1, \dots, G_k\}^+$.

Definition

The *upper-left-corner problem* or *hyperplane problem* for a matrix semigroup G asks for an algorithm which given a finite set of matrices $\{G_1, \dots, G_k\} \subseteq G$ determines whether $\exists M \in \{G_1, \dots, G_k\}^+$ with $M_{11} = 0$.

Lemma

The solvability of the mortality problem for $\mathrm{GL}_3(\mathbb{Z})$ implies the solvability of the upper left corner problem for $\mathrm{GL}_3(\mathbb{Z})$.

Theorem (Halava and Harju, 2001)

The upper-left-corner problem is undecidable for sets of 5 $3 \times 3-$integer matrices.

Note that this is for semigroups.

Theorem (Halava and Harju, 2001)

The upper-left-corner problem is undecidable for sets of 5 $3 \times 3-$integer matrices.

Note that this is for semigroups. What happens for groups?

Theorem (Halava and Harju, 2001)

The upper-left-corner problem is undecidable for sets of 5 $3 \times 3-$integer matrices.

Note that this is for semigroups. What happens for groups?

Theorem (Result from placement)

The upper-left-corner problem for groups is undecidable for sets of 17 $6 \times 6-$rational matrices.

This is achieved by reducing to the membership problem!

Free subgroups of SO_3

Motivated by study of Banach-Tarski paradox.

Free subgroups of SO_3

Motivated by study of Banach-Tarski paradox.

$$r_x^{\pm\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \mp\sin\alpha \\ 0 & \pm\sin\alpha & \cos\alpha \end{pmatrix} \qquad r_z^{\pm\alpha} = \begin{pmatrix} \cos\alpha & \mp\sin\alpha & 0 \\ \pm\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

are the rotations by an angle α around the x and z axis respectively. We ask for what α we have that $\langle r_x^\alpha, r_z^\alpha \rangle$ is a free group of rank 2.

Free subgroups of SO_3

A few results on this:

Free subgroups of SO_3

A few results on this:

Theorem (Result from placement)

If $\cos\alpha = \frac{2q}{q^2+1}$ *for* $q \in \mathbb{Q}, q \neq 0, 1$ *then the rotation matrices by* α *around the* x *and* z *axes generate a free group.*

Free subgroups of SO_3

A few results on this:

Theorem (Result from placement)

If $\cos\alpha = \frac{2q}{q^2+1}$ for $q \in \mathbb{Q}, q \neq 0, 1$ then the rotation matrices by α around the x and z axes generate a free group.

Theorem (Result from placement)

If $\cos\alpha$ is rational but not dyadic (i.e. $\neq \frac{m}{2^k}$), then the rotation matrices by α around the x and z axes generate a free group.

Free subgroups of SO_3

A few results on this:

Theorem (Result from placement)

If $\cos\alpha = \frac{2q}{q^2+1}$ for $q \in \mathbb{Q}, q \neq 0, 1$ then the rotation matrices by α around the x and z axes generate a free group.

Theorem (Result from placement)

If $\cos\alpha$ is rational but not dyadic (i.e. $\neq \frac{m}{2^k}$), then the rotation matrices by α around the x and z axes generate a free group.

Theorem (Swierczkowski, 1994)

If $\cos\alpha \in \mathbb{Q}\backslash\{0, \pm\frac{1}{2}, \pm1\}$, then the rotation matrices by α around the x and z axes generate a free group.