Braid Group Cryptography

Liam Hardiman

March 3, 2019

Finitely Presented Groups

A finitely presented group $G = \langle S | R \rangle$ is specified by two sets, $S = \{x_i\}_{i \in I}$ and $R = \{r_j\}_{j \in J}$.

Finitely Presented Groups

A finitely presented group $G = \langle S | R \rangle$ is specified by two sets, $S = \{x_i\}_{i \in I}$ and $R = \{r_j\}_{j \in J}$.

- S is a set of symbols called **generators**.

Finitely Presented Groups

A finitely presented group $G = \langle S|R \rangle$ is specified by two sets, $S = \{x_i\}_{i \in I}$ and $R = \{r_j\}_{j \in J}$.

- S is a set of symbols called **generators**.
- R is a set of words in S called **relators**. A **word** in S is a finite string consisting of symbols in S and the symbols x_i^{-1}, where $x_i \in S$. The empty string, e, is also a word.

Finitely Presented Groups

A finitely presented group $G = \langle S|R \rangle$ is specified by two sets, $S = \{x_i\}_{i \in I}$ and $R = \{r_j\}_{j \in J}$.

- S is a set of symbols called **generators**.
- R is a set of words in S called **relators**. A **word** in S is a finite string consisting of symbols in S and the symbols x_i^{-1}, where $x_i \in S$. The empty string, e, is also a word.
- We form a group by taking all possible words in S. The inverse of a word w is formed by writing the symbols in w in reverse order and replacing each x_j appearing in w by x_j^{-1}. The group operation is concatenation of words.

Finitely Presented Groups

We form G from S and R by taking all equivalence classes of words in S. Two words v and w are equivalent if v can be transformed into w by a finite sequence of these operations.

Finitely Presented Groups

We form G from S and R by taking all equivalence classes of words in S. Two words v and w are equivalent if v can be transformed into w by a finite sequence of these operations.

1. Replacing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with e

Finitely Presented Groups

We form G from S and R by taking all equivalence classes of words in S. Two words v and w are equivalent if v can be transformed into w by a finite sequence of these operations.

1. Replacing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with e
2. Inserting $x_i x_i^{-1}$ or $x_i^{-1} x_i$ at any position

Finitely Presented Groups

We form G from S and R by taking all equivalence classes of words in S. Two words v and w are equivalent if v can be transformed into w by a finite sequence of these operations.

1. Replacing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with e
2. Inserting $x_i x_i^{-1}$ or $x_i^{-1} x_i$ at any position
3. Replacing r_j with e

Finitely Presented Groups

We form G from S and R by taking all equivalence classes of words in S. Two words v and w are equivalent if v can be transformed into w by a finite sequence of these operations.

1. Replacing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with e
2. Inserting $x_i x_i^{-1}$ or $x_i^{-1} x_i$ at any position
3. Replacing r_j with e
4. Inserting r_j at any position

Finitely Presented Groups

We form G from S and R by taking all equivalence classes of words in S. Two words v and w are equivalent if v can be transformed into w by a finite sequence of these operations.

1. Replacing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with e
2. Inserting $x_i x_i^{-1}$ or $x_i^{-1} x_i$ at any position
3. Replacing r_j with e
4. Inserting r_j at any position

Equivalently, G is the quotient of the free group on S by the normal closure of R. We say G is **finitely presented** if S and R are finite sets.

Finitely Presented Groups

Some examples of finitely presented groups include...

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups
- The free group F_n on n generators

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups
- The free group F_n on n generators
- Finitely generated abelian groups

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups
- The free group F_n on n generators
- Finitely generated abelian groups
- The braid group B_n, $n \geq 0$.

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups
- The free group F_n on n generators
- Finitely generated abelian groups
- The braid group B_n, $n \geq 0$.

Nonexamples include

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups
- The free group F_n on n generators
- Finitely generated abelian groups
- The braid group B_n, $n \geq 0$.

Nonexamples include

- Any group with infinitely many generators, e.g. $\mathbb{Z}^{\oplus \mathbb{Z}}$

Finitely Presented Groups

Some examples of finitely presented groups include...

- Finite groups
- The free group F_n on n generators
- Finitely generated abelian groups
- The braid group B_n, $n \geq 0$.

Nonexamples include

- Any group with infinitely many generators, e.g. $\mathbb{Z}^{\oplus \mathbb{Z}}$
- There are finitely generated groups that are not finitely related, e.g. the wreath product of $\mathbb{Z}$ with itself.

The Word Problem

Say we have a finitely presented group G.

The Word Problem

Say we have a finitely presented group G.

The word problem in G

input: *two words v, w in the generators of G*

output: **yes** *if v is equivalent to w.* **no** *otherwise*

The Word Problem

Say we have a finitely presented group G.

The word problem in G

input: *two words v, w in the generators of G*

output: **yes** *if v is equivalent to w.* **no** *otherwise*

Example (The word problem in $F_2 = \langle a, b \rangle$)

Iteratively scan through both words, deleting adjacent inverses.

The Word Problem

Say we have a finitely presented group G.

The word problem in G

input: *two words v, w in the generators of G*

output: **yes** *if v is equivalent to w.* **no** *otherwise*

Example (The word problem in $F_2 = \langle a, b \rangle$)

Iteratively scan through both words, deleting adjacent inverses.
Given $v = aa^{-1}bba$ and $w = babb^{-1}a^{-1}ba$, we have

The Word Problem

Say we have a finitely presented group G.

The word problem in G

input: *two words v, w in the generators of G*
output: **yes** *if v is equivalent to w.* **no** *otherwise*

Example (The word problem in $F_2 = \langle a, b \rangle$)

Iteratively scan through both words, deleting adjacent inverses.
Given $v = aa^{-1}bba$ and $w = babb^{-1}a^{-1}ba$, we have

$$aa^{-1}bba = bba$$
$$babb^{-1}a^{-1}ba = bba.$$

The Word Problem

Say we have a finitely presented group G.

The word problem in G

input:	*two words v, w in the generators of G*
output:	**yes** *if v is equivalent to w.* **no** *otherwise*

Example (The word problem in $F_2 = \langle a, b \rangle$)

Iteratively scan through both words, deleting adjacent inverses.
Given $v = aa^{-1}bba$ and $w = babb^{-1}a^{-1}ba$, we have

$$aa^{-1}bba = bba$$
$$babb^{-1}a^{-1}ba = bba.$$

Output **yes**.

The Word Problem

In 1955 Pyotr Novikov showed that there are finitely presented groups in which the word problem is **undecidable** - it is provably impossible to construct an algorithm that always outputs the correct answer.

The Conjugacy Search Problem

Let G be a group.

The Conjugacy Search Problem

Let G be a group.

The Conjugacy Search Problem in G

input: *Two conjugate words u and v in the generators of G.*

output: *A word w such that $u = w^{-1}vw = v^w$*

The Conjugacy Search Problem

Let G be a group.

The Conjugacy Search Problem in G

input: *Two conjugate words u and v in the generators of G.*

output: *A word w such that $u = w^{-1}vw = v^w$*

This is analogous to the discrete logarithm problem in a finite abelian group H.

The Conjugacy Search Problem

Let G be a group.

The Conjugacy Search Problem in G

input: *Two conjugate words u and v in the generators of G.*
output: *A word w such that $u = w^{-1}vw = v^w$*

This is analogous to the discrete logarithm problem in a finite abelian group H.

Discrete Logarithm Problem in H

input: *Elements g, h of H such that $h \in \langle g \rangle$*
output: *An integer k such that $g^k = h$*

The Braid Group

Definition

The braid group on n strands, B_n is defined by the presentation

$$\begin{aligned} B_n = \langle \sigma_1, \sigma_2, \ldots, \sigma_{n-1} \mid \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j,\ |i-j| = 1; \\ \sigma_i\sigma_j = \sigma_j\sigma_i,\ |i-j| > 1\rangle. \end{aligned}$$

The Braid Group

Definition

The braid group on n strands, B_n is defined by the presentation

$$\begin{aligned} B_n = \langle \sigma_1, \sigma_2, \ldots, \sigma_{n-1} \mid \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j,\ |i-j| = 1; \\ \sigma_i\sigma_j = \sigma_j\sigma_i,\ |i-j| > 1\rangle. \end{aligned}$$

There is, however, a more geometric understanding of the braid group.

The Braid Group

- Arrange two sets of n items in vertical columns on opposite sides of the page. Fasten one end of a string to each item on the left side of the page. To each item on the right side attach the other end of one string. This connection is a **braid**.

The Braid Group

- Arrange two sets of n items in vertical columns on opposite sides of the page. Fasten one end of a string to each item on the left side of the page. To each item on the right side attach the other end of one string. This connection is a **braid**.
- The generator σ_i represents connecting the i-th item on the left to the $i + 1$st on the right and the $i + 1$st on the left to the i-th on the right with the latter string passing over the former.

The Braid Group

- Arrange two sets of n items in vertical columns on opposite sides of the page. Fasten one end of a string to each item on the left side of the page. To each item on the right side attach the other end of one string. This connection is a **braid**.
- The generator σ_i represents connecting the i-th item on the left to the $i+1$st on the right and the $i+1$st on the left to the i-th on the right with the latter string passing over the former.
- Two connections that can be made to look the same by tightening the strings are considered the same braid.

The Braid Group

- Arrange two sets of n items in vertical columns on opposite sides of the page. Fasten one end of a string to each item on the left side of the page. To each item on the right side attach the other end of one string. This connection is a **braid**.
- The generator σ_i represents connecting the i-th item on the left to the $i+1$st on the right and the $i+1$st on the left to the i-th on the right with the latter string passing over the former.
- Two connections that can be made to look the same by tightening the strings are considered the same braid.
- Composing two braids consists of drawing them next to one another, gluing the points in the middle, and connecting the strands.