Finitely Presented Groups

1. Basics on presentations, main problem.

2. Subgroups & quotient

3. Solving the word problem

1. Presentations

1.1 Basics

X finite set of generators

$\hat{X} = X \cup X^{-1}$

$\hat{X}^* =$ set of finite words in $\hat{X}$

includes $\varepsilon =$ the empty words

$R \subseteq \hat{X}^*$ finite set of relators

Define $\sim$ as the finest equivalence relation on $\hat{X}$ with

- $u x x^{-1} v \sim uv \quad \forall u,v \in \hat{X}^*, \forall x \in \hat{X}$
- $urv \sim uv \quad \forall r \in R$

Set $G = \hat{X}^*/\sim$ is a group with concatenation as product.

We call G the finitely presented group with generators X and relations R:

$$G = \langle X \mid R \rangle .$$

There is a natural way to map

$$X^* \to G , \quad w \mapsto \overline{w} .$$

<u>examples</u>:

$\langle C \mid C^k \rangle$ cyclic group of order k.

$\langle C \mid \emptyset \rangle \cong \mathbb{Z}$

$\langle A, B \mid \underbrace{ABAB^{-1}}_{\Leftrightarrow AB = BA} \rangle \cong \mathbb{Z}^2$

$\langle P, Q \mid P^n, Q^2, (PQ)^2 \rangle \cong D_{2n}$ dihedral group

$\langle S, T \mid S^3, T^2 \rangle \cong PSL_2(\mathbb{Z})$

lol trying to spell "Bilbao" but you can't repeat letters

$\rightarrow \langle B, I, L, O \mid$ BILL
OIL
BILBO
BOIL
OIL
LIB
... $\rangle$ = trivial

after enough words

Thm (Dyck 1882)

Let $G = \langle X \mid R \rangle$. Let H be any group and $f: X \to H$.

Set $f(x^{-1}) = f(x)^{-1}$.

If
$$f(x_1) f(x_2) \cdots f(x_n) =_H 1$$
whenever
$$x_1, x_2, \ldots, x_n \in R,$$
then there is a unique group hom.
$$\hat{f} : G \longrightarrow H$$
st $\hat{f}(x) = f(x) \quad \forall x \in X.$

This means whenever a group H can be generated by X and fulfills the relations R. It is a quotient of
$$G = \langle X \mid R \rangle.$$
"universal group G"

ex: $\langle A, B \mid \emptyset \rangle$ free groups on 2 gens

all 2-gen groups are quotients.

ex: $\langle C \mid C^k \rangle$ the quotients are cyclic groups of orders dividing k.

1.2 Three fundamental problems

(by Dehn 1911)

① For $G = \langle X \mid R \rangle$ and $w \in \hat{X}$, decide whether:

by ALGORITHM

$$\overline{w} =_G 1. \quad \text{word problem}$$

② For $G = \langle X \mid R \rangle$ and $v, w \in \hat{X}^*$ decide if there is a word

$u \in \hat{X}^*$ st

$$w =_G uvu^{-1} \quad \text{conjugacy problem}$$

③ For $G = \langle X \mid R \rangle$ and
$H = \langle Y \mid S \rangle$

decide if $G \cong H$. isomorphism problem.

or even decide if G is trivial.

In general, these problems are "hard".

GAP demo: Set $G := \langle s, t \mid s^2, t^3 \rangle$.
ASK order $(G.1)$

took 8s to figure out order of first gen.

we will see what GAP is trying to do.

Thm (Markov-Post 1947, Novikov 1955)

There is a finite presentation $G = \langle X \mid R \rangle$ for which there is no algorithm to solve the word prob.

Proof sketch:

There is a subset S of $\hat{X}^*$ that has the remarkable property that:

← tricky, go back to math foundations

- There is a Turing Machine (TM) that halts on S.
- There is no TM that halts on S^c.

(The formal language of TM's, analogous to regular lang. of finite state automata).

To every TM, associate a semi-group approximating it.

Take T^* to be the TM that halts on S, we get a semigroup G

$S = L$.

$$\{ w \in X^* \mid T^* \text{ halts on } w \} = S$$

$$\parallel$$

$$\{ w \in \hat{X}^* \mid \overline{w} =_G 1 \}$$

So there is an algorithm that stops on words that are trivial in G.

The problem is that there is no algorithm/TM that stops on non-trivial elements.

Remark: • There are of course, particular classes of groups for which we <u>can</u> solve the word prob.
(ex: abelian grps, polycyclic grps).

• We will look at some <u>procedures</u> that may work

NOT algorithm.
A procedure is not bound to stop.

for some particular groups.

this is the best we can hope for due to them.

GAP demo:

• If you define f:=FreeGroup(p,q).

r:=Relations (*)

g:= f/r.

p $\neq$ g.1 even if they are rep'ed by same string.

• Sometimes GAP can recognize that the size of G is infinite

→ G was abelian

take the abelianization.

GAP was not able to recognize order of element.

1.3 Transforming presentations

Begin w/ f.g group, $G = \langle X \mid R \rangle$.

Q: What kind of things can we do with presentation so we don't change the underlying group?

Tietze transformations

① Add relators: if they follow from others.

② Remove relators: if they are redundant.

③ Add generator and a relator
if a word appears frequently as a subword, might be practical to add a generator x, and a relator xw^{-1}

where $x \equiv_G w$.

ex: $G = \langle a, b \mid a^2, aba^3ba^3b, (ab)^6, ab^5 \rangle$

③ $a^3 = c$

$\langle a, b, c \mid a^2, a(bc)^2b, (ab)^6, ab^5, ca^{-3} \rangle$

① $ca^{-3} \cdot a^2 = ca^{-1}$

$\langle a, b, c \mid a^2, a(bc)^2b, (ab)^6, ab^5, ca^{-3}, ca^{-1} \rangle$

④

$\langle a, b \mid a^2, a\underbrace{(ba)^2b}_{(ab)^3}, (ab)^6, ab^5, a^{-2} \rangle$

...

$\langle b \mid b^2 \rangle = \mathbb{Z}/2\mathbb{Z}$.

Thm (Tietze)

Given two finite presentations of the same group, one can be obtained from the other by a seq. of Tietze transformation.

GAP demo:

- Tz Print ___ fn's.
- Add Generator, Add Relator, Remove Relator.

FASTER TzSubstitute(p, ab)

add new gen (ab) to pres. and substitute it in

EVEN FASTER TzGo(p)

does Tietze transfo on its own to make presentation shorter.

Simplifying presentation makes computation faster.

```
# Tietze transformations on a given presentation: MANUALLY
f := FreeGroup("a","b");
rels := [];
g := f/ParseRelators(f,"1=a^2=b^3=(ab)^11=[a,b]^6=(ababab^-1)^6");
Size(g); time;
p := PresentationFpGroup(g);
GeneratorsOfPresentation(p);
TzPrintGenerators(p);
TzPrintRelators(p);
TzPrintPresentation(p);
TzPrintPairs(p); # find often occuring words of length 2
AddGenerator(p);
TzPrintPresentation(p);
GeneratorsOfPresentation(p);
a := GeneratorsOfPresentation(p)[1];
b := GeneratorsOfPresentation(p)[2];
c := GeneratorsOfPresentation(p)[3];
AddRelator(p,a*b*c^-1);
TzPrintPresentation(p);
AddRelator(p,c^11);
RemoveRelator(p,3);
TzPrintPresentation(p);
```

```
# SEMI-MANUALLY
p := PresentationFpGroup(g);
TzPrintPresentation(p);
a := GeneratorsOfPresentation(p)[1];
b := GeneratorsOfPresentation(p)[2];
TzSubstitute(p,a*b);
TzPrintPresentation(p);
TzOptions(p).printLevel := 2;
TzGo(p);
TzPrintPresentation(p);
TzPrintPairs(p);
c := GeneratorsOfPresentation(p)[2];
TzSubstitute(p,b*c);
TzGo(p);
TzPrintPresentation(p);
TzPrintPairs(p);
SimplifyPresentation(p); # synonym for TzGo
TzPrintPresentation(p);
h := FpGroupPresentation(p);
Size(h); time; # 2x faster than with the old presentation
```

this leads to

Efficient Presentations

$G = \langle x \mid R \rangle$

$\delta(x \mid R) = |x| - |R|$ deficiency of presentation

ex: $\langle c \mid c^{k} \rangle$ $\delta = 1 - 1 = 0$

ex (from Tietze):

original: $\delta = 2 - 4 = -2$

after Tietze $\delta = 1 - 1 = 0$.

We can always add relators that follow from others

$\Rightarrow$ can always decrease the deficiency.

Q: What is the largest possible δ?

Deficiency of G $= \max\limits_{\substack{\delta(x \mid R) \\ G \cong \langle x \mid R \rangle}} \delta$

Thm A group w/ $\delta > 0$ is infinite.
(ie all finite groups have $\delta \leq 0$.)

Thm
when $\delta(G) \geq 2$, then G has a subgroup H of finite index with

$H \twoheadrightarrow F_2$.

NOTE: G fin. pres.

$\delta(G) < \infty$

Since $\delta(G) \leq \underbrace{d(G)}_{\text{smallest \# of generators.}}$

A better bound:

Thm: G finite

$$\delta(G) \leq -\operatorname{rank} \underbrace{H_2(G, \mathbb{Z})}_{\text{Schur multiplier of } G.}$$

If we have equality we say that

G is efficient.