Computing with finitely presented groups

Heiko Dietrich

School of Mathematics
Monash University, Australia

3-7 June 2024
IMS NSU

▸ Go to Overview

Welcome!

Your teaching team

Lectures will be delivered by **me** . . .
Problem sessions are supported by **Khánh Lê** (Rice University).

In this lecture series we talk about some

computational aspects of finitely presented groups.

Assumed knowledge

Some basis group theory (free groups, group actions, cosets, . . .)
I'm happy to (try to) answer any questions . . .

Lecture material

We cover some fundamental concepts in CGT. The slides are quite dense, but I'll try to be slow . . . The material will be online at

`users.monash.edu/~heikod/imsnus2024`.

Outline

- ▸ Go to Lecture 1

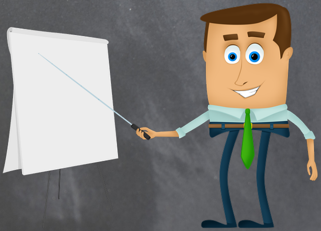
  group presentations, Tietze transformations, von Dyck, coset enumeration, . . .

- ▸ Go to Lecture 2

  polycyclic groups, quotient algorithms, . . .

- ▸ Go to Lecture 3

  rewriting systems, automatic groups, . . .



Main resources

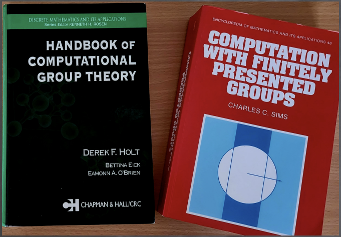
Recommended reading:

- **Handbook of computational group theory** (Holt, Eick, O'Brien)
- **Computation with finitely presented groups** (Sims)
- **Presentations of groups** (Johnson)

See the website for more references ...

Special thanks to:

- Alexander Hulpke: notes http://tinyurl.com/ymmamzv7
- Derek Holt: slides http://tinyurl.com/4va9vt54
- Bettina Eick, Murray Elder, Eamonn O'Brien


DISCRETE MATHEMATICS AND ITS APPLICATIONS
Series Editor KENNETH H. ROSEN
HANDBOOK OF COMPUTATIONAL GROUP THEORY
DEREK F. HOLT
BETTINA EICK
EAMONN A. O'BRIEN
CHAPMAN & HALL/CRC
ENCYCLOPEDIA OF MATHEMATICS AND ITS APPLICATIONS 48
COMPUTATION WITH FINITELY PRESENTED GROUPS
CHARLES C. SIMS

First Lecture

▸ Go to Overview

▸ Go to Quotient Algorithm

Computing with groups

In what format is the group given?

How well we can compute with a group depends heavily on how the group is represented. For example, the dihedral group D_8 can be defined as a $\mathrm{Sym}(\square)$, or as a...

- ... permutation group

$$\langle (1,2,3,4), (1,3) \rangle;$$

- ... matrix group

$$\langle \left(\begin{smallmatrix} 0 & 1 \\ 2 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right) \rangle \leqslant \mathrm{GL}_2(3);$$

- ... finitely presented group

$$\langle r, m \mid r^4, m^2, rmrm \rangle.$$

Group presentations describe groups in a compact way. They are the objects of interest in combinatorial/geometric group theory, and occur naturally in areas such as topology. Von Dyck (1882) did the first systematic study of groups given by generators and relations.

Recall: free groups

Free group

Let $X \neq \emptyset$ be a set that is embedded in a group F via $\iota\colon X \hookrightarrow F$. Then F **is free on** X if every map $\varphi\colon X \to H$ into a group H extends to a *unique* homomorphism $\psi\colon F \to H$, that is, $\psi \circ \iota = \varphi$.

$$\begin{tikzcd} & F \arrow[d, "\exists!\psi"] \\ X \arrow[ur, "\iota"] \arrow[r, "\forall\varphi"] & H \end{tikzcd}$$

E.g.: $(\mathbb{Z}, +)$ is free on $\{1\}$; the group $\langle \left(\begin{smallmatrix} 1 & 0 \\ 2 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\right) \rangle$ is free on its generating set.

Fact: Up to isomorphism, there is a unique group F_X that is free on X.

- Let S^* be the set of all words over $S = X \sqcup X^{-1}$ (*disjoint copy; symbols!*)
- Words $w, w' \in S^*$ are *equivalent* if w' is constructed from w by deleting or inserting subwords xx^{-1} or $x^{-1}x$; write $[w]$ for the equivalence class.
- Let $F_X = \{[w] : w \in S^*\}$ with $[w][w'] = [ww']$ and $\iota\colon X \to F_X$, $x \mapsto [x]$.

E.g.: If $X = \{a, b\}$, then $S = \{a, b, a^{-1}, b^{-1}\}$ (*symbols!*) and

$$S^* = \{\varepsilon, a, b, a^{-1}, b^{-1}, aa, ab, aa^{-1}, ab^{-1}, ba, bb, \ldots\}.$$

We have $baa^{-1}a \neq ba$ as words, but $[baa^{-1}a] = [ba]$ in F_X.

Recall: group presentations

Group presentation

Let X be a non-empty set and $\mathcal{R} \subseteq F_X$.
The group defined by the *presentation* $\{X \mid \mathcal{R}\}$ is

$$\langle X \mid \mathcal{R}\rangle = F_X/R \quad \text{where} \quad R = \mathcal{R}^{F_X} \text{ (normal closure).}$$

The sets X and $\mathcal{R}$ are the *generators* and *defining relators* of the presentation.

It's a bit clumsy: The elements of $\langle X \mid \mathcal{R}\rangle$ with $X = \{x\}$ and $\mathcal{R} = \{[x]^3\}$ are

$$[\varepsilon]\langle [x]^3\rangle^{F_{\{x\}}}, \qquad [x]\langle [x]^3\rangle^{F_{\{x\}}}, \qquad [x]^2\langle [x]^3\rangle^{F_{\{x\}}}.$$

Better: Let's write these elements as $1, x, x^2$, and also $G = \langle x \mid x^3\rangle = \{1, x, x^2\}$.

Convention: Let $G = \langle X \mid \mathcal{R}\rangle = F_X/R$. We often identify $w \in S(X)^*$ with

$$\underbrace{w}_{\in S(X)^*} = \underbrace{[w]}_{\in F_X} = \underbrace{[w]R}_{\in G}.$$

Careful: $x^4 \neq_{S(X)^*} x^5x^{-1} =_{F_X} x^4$ and $x^4 \neq_{F_X} x =_G x^4$.

Recall: group presentations

Example 1

Let $X = \{r, m\}$ and $\mathcal{R} = \{r^4, m^2, \overbrace{rmrm}^{\text{relator}}\}$, and

$$G = \langle X \mid \mathcal{R} \rangle = \langle r, m \mid r^4, m^2, \underbrace{rm = mr^3}_{\text{relation}} \rangle.$$

What can we say about G?

- As elements in G, we have $r^4 = 1$, $m^2 = 1$, and $rmrm = 1$; the latter is equivalent to $rm = m^{-1}r^{-1} = mr^3$.
- This can be used to rewrite elements, e.g., move all m's to the left:

$$m\underline{rm}mrr = mm\underline{r^3m}rr = mmmr^9rr = m^3r^{11} = mr^3.$$

- Thus, $G = \{m^i r^j \mid i = 0, 1 \text{ and } j = 0, 1, 2, 3\}$, and so $|G| \leqslant 8$.
- Later: $|G| = 8$ and $G \cong \langle M, R \rangle = D_8$ for $R = (1, 2, 3, 4)$ and $M = (1, 3)$.

Some examples in GAP

GAP: Groups, Algorithms, and Programming (www.gap-system.org)

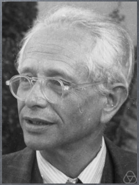
```
gap> F := FreeGroup(["r","m"]);;
gap> R := [F.1^2, F.2^4, F.2^F.1/F.2^-1];;
gap> G := F/R;
<fp group on the generators [ r, m ]>
gap> StructureDescription(G);
"D8"
```

```
gap> H   := Group([(1,2,3,4),(1,3)]);;
gap> Hfp := Image(IsomorphismFpGroup(H));
<fp group on the generators [ F1, F2, F3 ]>
gap> RelatorsOfFpGroup(Hfp);
[ F1^2, F1^-1*F2*F1*F2^-1*F3^-1, F2^2*F3^-1,
  F1^-1*F3*F1*F3^-1, F2^-1*F3*F2*F3^-1, F3^2 ]
gap> Size(Hfp);
8
```

```
gap> F := FreeGroup(["a","b"]);;
gap> W := F/ParseRelators(F,"a^3*b^4*a^5*b^7, a^2*b^3*a^7*b^8");;
gap> # Challenging -- later more!
```

("Wicks' Group", see Havas, Havas, Kenne, Rees, *Some challenging group presentations*, 1999)

Dehn problems (1911)



In the following we focus on finitely presented groups (**fp groups**), that is, $G = \langle X \mid \mathcal{R} \rangle$ with $|X|$ and $|\mathcal{R}|$ finite. **Dehn** (1910) asked: Given an fp group G, is there an algorithm that

- decides whether a word in the generators represents 1_G? (**Word Problem**)
- decides whether two elements in G are conjugate? (**Conjugacy Problem**)
- decides whether G is isomorphic to an fp group H? (**Isomorphism Problem**)

Novikov 1955, Boone 1954-57, Britton 1958

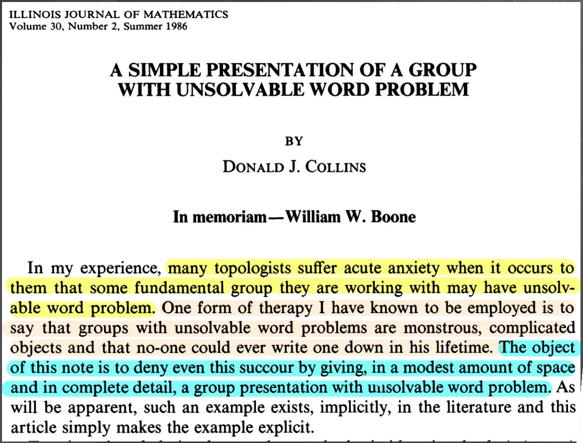
There is a finite presentation $G = \langle X \mid \mathcal{R} \rangle$ for which there is no algorithm that, given two words u and v over $X \cup X^{-1}$, decides whether $u = v$ in G.

See also Miller (1992) "*Decision problems for groups (...)*" for a survey and detailed discusssion.

Seems hopeless...? The aim of these lectures is to discuss approaches that work.

Also, maybe these presentations are *rare*...?

They exist...



ILLINOIS JOURNAL OF MATHEMATICS
Volume 30, Number 2, Summer 1986

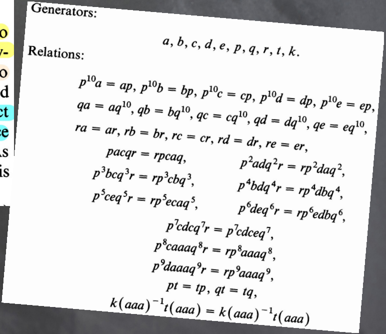
A SIMPLE PRESENTATION OF A GROUP WITH UNSOLVABLE WORD PROBLEM

BY

DONALD J. COLLINS

In memoriam—William W. Boone

In my experience, many topologists suffer acute anxiety when it occurs to them that some fundamental group they are working with may have unsolvable word problem. One form of therapy I have known to be employed is to say that groups with unsolvable word problems are monstrous, complicated objects and that no-one could ever write one down in his lifetime. The object of this note is to deny even this succour by giving, in a modest amount of space and in complete detail, a group presentation with unsolvable word problem. As will be apparent, such an example exists, implicitly, in the literature and this article simply makes the example explicit.



Generators:

$$a, b, c, d, e, p, q, r, t, k.$$

Relations:

$$p^{10}a = ap,\ p^{10}b = bp,\ p^{10}c = cp,\ p^{10}d = dp,\ p^{10}e = ep,$$
$$qa = aq^{10},\ qb = bq^{10},\ qc = cq^{10},\ qd = dq^{10},\ qe = eq^{10},$$
$$ra = ar,\ rb = br,\ rc = cr,\ rd = dr,\ re = er,$$
$$pacqr = rpcaq, \qquad p^2adq^2r = rp^2daq^2,$$
$$p^3bcq^3r = rp^3cbq^3, \qquad p^4bdq^4r = rp^4dbq^4,$$
$$p^5ceq^5r = rp^5ecaq^5, \qquad p^6deq^6r = rp^6edbq^6,$$
$$p^7cdcq^7r = p^7cdceq^7,$$
$$p^8caaaq^8r = rp^8aaaq^8,$$
$$p^9daaaq^9r = rp^9aaaq^9,$$
$$pt = tp,\ qt = tq,$$
$$k(aaa)^{-1}t(aaa) = k(aaa)^{-1}t(aaa)$$

So now let's see what *can* be done...

Tietze transformation

Tietze transformations modify presentations $G = \langle X \mid \mathcal{R}\rangle = F_X/\mathcal{R}^{F_X}$ without changing the isomorphism type of the group:

- **Adjoint a relator:** If $r \in \mathcal{R}^{F_X}$, then $G = \langle X \mid \mathcal{R} \cup \{r\}\rangle$.
  E.g.: $\langle g, h \mid ghg = hgh\rangle = \langle g, h \mid ghg = hgh, hgh\underline{ghg} = hgh\underline{hgh}\rangle$.

- **Remove a relator:** If $r \in \mathcal{R}$ lies in $(\mathcal{R} \setminus \{r\})^{F_X}$, then $G = \langle X \mid \mathcal{R} \setminus \{r\}\rangle$.
  E.g.: $\langle g, h \mid ghg = hgh, hghghg = hghhgh\rangle = \langle g, h \mid hghghg = hghhgh\rangle$.

- **Adjoin a generator:**
  If $t \notin X \cup X^{-1}$ and $w \in F_X$, then $G \cong \langle X \cup \{t\} \mid R \cup \{t = w\}\rangle$.
  E.g.: $\langle g, h \mid \underline{hg}\,\underline{hg}\,\underline{hg} = \underline{hgh}\,\underline{hgh}\rangle \cong \langle g, h, s, t \mid t = hg, s = th, t^3 = s^2\rangle$.

- **Remove a generator:**
  If $x \in X$ occurs in $\mathcal{R}$ only once, say in r, then $G \cong \langle X \setminus \{x\} \mid \mathcal{R} \setminus \{r\}\rangle$.
  E.g.: $\langle g, h, s, t \mid t = hg, s = th, t^3 = s^2\rangle \cong \langle s, t \mid t^3 = s^2\rangle$.

GAP

GAP's command `SimplifiedFpGroup` applies Tietze transformations, aiming to produce a "simpler" presentation.

```
gap> F := FreeGroup(["g","h","s","t"]);;
gap> G := F / ParseRelators(F, "t=h*g, s=t*h, t^3=s^2");
<fp group of size infinity on the generators [ g, h, s, t ]>
gap> Gnew := SimplifiedFpGroup(G);
<fp group of size infinity on the generators [ g, t ]>
gap> RelatorsOfFpGroup(Gnew);
[ g*t*g*t^-2 ]
# new relation is t^2=gtg
# original relation was t=hg, so t^2=gtg if and only if hgh=ghg
gap> iso := IsomorphismSimplifiedFpGroup(G);
[ g, h, s, t ] -> [ g, t*g^-1, g*t, t ]
gap>Image(iso)=Gnew;
true
```

Tietze transformation

Another example:

Example 2

The group $G = \langle g, h \mid ghghg \rangle$ is isomorphic to $\langle a \mid \emptyset \rangle \cong (\mathbb{Z}, +)$.

Interesting:

Tietze transformations determine isomorphism classes

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

A proof can be found in Johnson §4.4.

Next: von Dyck's Theorem!

A crucial tool: von Dyck's Theorem

Let $G = \langle X \mid \mathcal{R} \rangle$ be an fp group with $X = \{x_1, \ldots, x_n\}$, so elements in G are represented by words $x_{i_1}^{e_1} \cdots x_{i_k}^{e_k}$ with $k \geqslant 0$, each $x_{i_j} \in X$ and $e_j \in \{\pm 1\}$.

Despite undecidability issues, there is an easy criterion that helps us to answer: *When does a map from X into some group H extend to a homomorphism?* This is arguably one of the most important tools for working with fp groups.

Von Dyck's Theorem

Let $G = \langle X \mid \mathcal{R} \rangle$ be as above and let $\varphi\colon X \to H$ be a map into some group H. The map φ extends to a (unique) group homomorphism $G \to H$ if and only if every $x_{i_1}^{e_1} \cdots x_{i_k}^{e_k} \in \mathcal{R}$ satisfies $\varphi(x_{i_1})^{e_1} \cdots \varphi(x_{i_k})^{e_k} = 1_H$.

Proof. If φ extends to a hom. and $r \in \mathcal{R}$, then $r =_G 1_G$ is mapped to 1_H. Conversely, let $\psi\colon F_X \to H$ be the unique hom. with each $\psi(x_i) = \varphi(x_i)$. By assumption, $\mathcal{R} \leqslant \ker \psi$, and so also $R = \mathcal{R}^{F_X} \leqslant \ker \psi$. Thus, ψ induces a well-defined hom. $F_X/R \to H$, $wR \mapsto \psi(w)$. The claim follows since $G = F_X/R$.

So: $G = \langle X \mid \mathcal{R} \rangle$ is the "largest" group whose generating set X satisfies $\mathcal{R}$.

Example: von Dyck's Theorem

Example 3

Show that $G = \langle r, m \mid m^2, r^4, mrmr \rangle \cong D_8$; let D_8 be given as a permutation group generated by $M = (1,3)$ and $R = (1,2,3,4)$.

Guess the map $\varphi\colon \{r,m\} \to D_8$ with $\varphi(m) = M$ and $\varphi(r) = R$; check relators:

$$\begin{aligned} m^2 &: \varphi(m)^2 = M^2 = (), \quad ✓ \\ r^4 &: \varphi(r)^4 = R^4 = (), \quad ✓ \\ rmrm &: \varphi(r)\varphi(m)\varphi(r)\varphi(m) = RMRM = (). \quad ✓ \end{aligned}$$

Hence, von Dyck's Theorem shows that φ induces a homomorphism $G \to D_8$. Since its image contains generators of D_8, it is an epimorphism, so $|G| \geqslant 8$. We knew already that $|G| \leqslant 8$, so $|G| = 8$ and $\varphi\colon G \to D_8$ is an isomorphism.

Example 4

The Wicks group $W = \langle a, b \mid a^3b^4a^5b^7, a^2b^3a^7b^8 \rangle$ has C_{11} as a quotient.

We'll see von Dyck's Theorem again later.

Next up: Coset enumeration – another fundamental method for fp groups.

Coset actions

Recall that G acts on a set Ω if there is a homomorphism $\varphi\colon G \to \mathrm{Sym}(\Omega)$.
If $g \in G$ and $\omega \in \Omega$, then we usually write $\omega^g = \omega^{\varphi(g)}$.

The stabiliser of ω in G is the subgroup $G_\omega = \{g \in G : \omega^g = \omega\}$;
the orbit of ω is $\omega^G = \{\omega^g : g \in G\}$. The action is transitive if $\Omega = \omega^G$.

Orbit-Stabiliser Theorem (OST)
If G acts on Ω, then $G_\omega\backslash G \to \omega^G$, $G_\omega g \mapsto \omega^g$, is a bijection for every $\omega \in \Omega$.

Coset action. Let $H \leqslant G$ be a subgroup of finite index n. Then G acts transitively via right multiplication on the set of all right cosets

$$H\backslash G = \{Hg : g \in G\};$$

specifically, $k \in G$ maps Hg to $H(gk)$.

Note: The OST defines a **bijection** between the conjugacy classes of index-n subgroups of G and the *equivalence classes* of transitive G-actions on n points.

Coset enumeration

Given: fp group $G = \langle X \mid \mathcal{R}\rangle$ and subgroup $U = \langle Y\rangle$ for some finite $Y \subseteq G$.

Want: Determine the index $[G:U]$, that is, the size of $U\backslash G$.

Idea: Consider the G-action on cosets $U\backslash G$ and enumerate the orbit of U.

Wait... Testing $r =_G s$ is undecidable in general; how to test $Ur = Us$?

Simple observation: *It feels a bit like magic* – but if $[G:U]$ happens to be finite, then one can attempt computing it based on the following observations:

- $Uu = U$ for every generator $u \in Y$ of U
- $Ugr = Ug$ for every coset Ug and defining relator $r \in \mathcal{R}$

Approach: Label cosets $1, 2, \ldots$ (with U labelled "1") and act with generators. E.g., if $X = \{a, b, c, d\}$ and $Y = \{cdb\}$, then $Ucdb = U$, and so

$$\underbrace{1}_{U} \xrightarrow{c} \underbrace{2}_{Uc} \xrightarrow{d} \underbrace{3}_{Ucd} \xrightarrow{b} \underbrace{1}_{Ucdb=U}.$$

We don't know whether $1, 2, 3$ are distinct, but we know that cdb maps 1 to 1.

Coset enumeration

We do this by attempting to complete three sets of tables.

Coset table. Here we collect how generators act on cosets.
Row labels: cosets $1, 2, \ldots$. **Column labels:** x and x^{-1} for $x \in X$.
Entry b in row a and column y if $a^y = b$; i.e. if y maps coset a to coset b.

Relator tables. For each defining relator $x_{i_1}^{e_1} \ldots x_{i_k}^{e_k} \in \mathcal{R}$ we have one table:
Row labels: cosets $1, 2, \ldots$. **Column labels:** $r_j = x_{i_1}^{e_i} \ldots x_{i_j}^{e_j}$ for $j = 1, \ldots, k$.
Entry b in row a and column r_j if $a^{r_j} = b$; row a and last column has entry a.

Subgroup tables. For each generator $x_{i_1}^{e_1} \ldots x_{i_k}^{e_k}$ of U we have one table:
Row labels: single row 1. **Column labels:** $y_j = x_{i_1}^{e_i} \ldots x_{i_j}{}^{e_j}$ for $j = 1, \ldots, k$.
Entry b in column y_j if $1^{y_j} = b$; last column has entry 1.

Fill these tables: Put a label (not previously used) into an empty spot, *draw conclusions, and resolve collisions* ... Finish when all tables have their first m rows filled with numbers $1, \ldots, m$. Then $[G : U] = m$ is established.

Coset enumeration example: $G = \langle a, b \mid ab^{-1}ab, b^2\rangle$ and $U = \langle a^2\rangle \leqslant G$.

C-table

	name	a	b	a^{-1}	b^{-1}
1	U	2		2	
2	Ua	1		1	

R-tables

	a	$\overset{b^{-1}}{\curvearrowright}$	ab^{-1}	$\overset{a}{\curvearrowright}$	$ab^{-1}a$	$\overset{b}{\curvearrowright}$	$ab^{-1}ab$
1	2						1
2	1						2

	b	$\overset{b}{\curvearrowright}$	b^2
1			1
2			2

S-table

	a	$\overset{a}{\curvearrowright}$	a^2
1	2		1

Notes

1) Pick empty spot: 2

2) Consequences:

- coset 2 is Ua
- if $1^a = 2$, then $2^{a^{-1}} = 1$
- update other tables

3) Deductions:

- S-table forces $2^a = 1$
- update other tables

4) All done? Go to 1)

Coset enumeration example: $G = \langle a, b \mid ab^{-1}ab, b^2 \rangle$ and $U = \langle a^2 \rangle \leqslant G$.

C-table

	name	a	b	a^{-1}	b^{-1}
1	U	2		2	
2	Ua	1		1	3
3	Uab^{-1}		2		

R-tables

	a	$\overset{b^{-1}}{\curvearrowright} ab^{-1}$	$\overset{a}{\curvearrowright} ab^{-1}a$	$\overset{b}{\curvearrowright} ab^{-1}ab$
1	2	3		1
2	1			2
3				3

	b	$\overset{b}{\curvearrowright} b^2$
1		1
2		2
3		3

S-table

	a	$\overset{a}{\curvearrowright} a^2$
1	2	1

Notes

1) Pick empty spot: 3

2) Consequences:

- coset 3 is Uab^{-1}
- if $2^{b^{-1}} = 3$, then $3^b = 2$
- update other tables

3) All done? Go to 1)
(*Actually: not all done!*)

Coset enumeration example: $G = \langle a, b \mid ab^{-1}ab, b^2 \rangle$ and $U = \langle a^2 \rangle \leqslant G$.

C-table

	name	a	b	a^{-1}	b^{-1}
1	U	2		2	
2	Ua	1	4	1	3
3	Uab^{-1}		2		
4	Uab				2

R-tables

	a	$\overset{b^{-1}}{\curvearrowright} ab^{-1}$	$\overset{a}{\curvearrowright} ab^{-1}a$	$\overset{b}{\curvearrowright} ab^{-1}ab$
1	2	3		1
2	1			2
3				3
4				4

	b	$\overset{b}{\curvearrowright} b^2$
1		1
2	4	2
3		3
4		4

S-table

	a	$\overset{a}{\curvearrowright} a^2$
1	2	1

Notes

1) Pick empty spot: 4

2) Consequences:

- coset 4 is Uab
- if $2^b = 4$, then $4^{b^{-1}} = 2$
- update other tables

3) Deductions:

- R-table forces $4^b = 2$
- so $2^{b^{-1}} = 4$, which is a *collision* with $2^{b^{-1}} = 3$
- update tables with $3 = 4$

Coset enumeration example: $G = \langle a, b \mid ab^{-1}ab, b^2 \rangle$ and $U = \langle a^2 \rangle \leqslant G$.

C-table

	name	a	b	a^{-1}	b^{-1}
1	U	2		2	
2	Ua	1	3	1	3
3	Uab^{-1}		2		2

R-tables

	a	$\overset{b^{-1}}{\curvearrowright} ab^{-1}$	$\overset{a}{\curvearrowright} ab^{-1}a$	$\overset{b}{\curvearrowright} ab^{-1}ab$
1	2	3		1
2	1			2
3				3

	b	$\overset{b}{\curvearrowright} b^2$
1		1
2	3	2
3	2	3

S-table

	a	$\overset{a}{\curvearrowright} a^2$
1	2	1

Notes

1) Pick empty spot: 4

2) Consequences:

- coset 4 is Uab
- if $2^b = 4$, then $4^{b^{-1}} = 2$
- update other tables

3) Deductions:

- R-table forces $4^b = 2$
- so $2^{b^{-1}} = 4$, which is a *collision* with $2^{b^{-1}} = 3$
- update tables with $3 = 4$
- also update $3^b = 2$

4) All done? Go to 1)

Coset enumeration example: $G = \langle a, b \mid ab^{-1}ab, b^2 \rangle$ and $U = \langle a^2 \rangle \leqslant G$.

C-table

	name	a	b	a^{-1}	b^{-1}
1	U	2	4	2	4
2	Ua	1	3	1	3
3	Uab^{-1}	4	2	4	2
4	Ub	3	1	3	1

R-tables

	a	$\overset{b^{-1}}{\curvearrowright} ab^{-1}$	$\overset{a}{\curvearrowright} ab^{-1}a$	$\overset{b}{\curvearrowright} ab^{-1}ab$
1	2	3	4	1
2	1	4	3	2
3	4	1	2	3
4	3	2	1	4

	b	$\overset{b}{\curvearrowright} b^2$
1	4	1
2	3	2
3	2	3
4	1	4

S-table

	a	$\overset{a}{\curvearrowright} a^2$
1	2	1

Notes

1) Pick empty spot: 4

2) Consequences:

- if $1^b = 4$, then $4^{b^{-1}} = 1$
- update other tables

3) Deductions:

- R-table forces $4^b = 1$
- update tables

4) Shortcut:

- We know b's action; work backwards!
- Now we see $3^a = 4$
- Update rest...

Coset enumeration example: $G = \langle a, b \mid ab^{-1}ab, b^2 \rangle$ and $U = \langle a^2 \rangle \leqslant G$.

C-table

	name	a	b	a^{-1}	b^{-1}
1	U	2	4	2	4
2	Ua	1	3	1	3
3	Uab^{-1}	4	2	4	2
4	Ub	3	1	3	1

Induced perms:

$a \mapsto \alpha = (1,2)(3,4)$

$b \mapsto \beta = (1,4)(2,3)$

R-tables

	a	$\overset{b^{-1}}{\curvearrowright} ab^{-1}$	$\overset{a}{\curvearrowright} ab^{-1}a$	$\overset{b}{\curvearrowright} ab^{-1}ab$
1	2	3	4	1
2	1	4	3	2
3	4	1	2	3
4	3	2	1	4

	b	$\overset{b}{\curvearrowright} b^2$
1	4	1
2	3	2
3	2	3
4	1	4

S-table

	a	$\overset{a}{\curvearrowright} a^2$
1	2	1

Notes

- Tables are *complete*.
- C-table columns define permutations of $\{1,2,3,4\}$ and a^{-1}, b^{-1} act as inverses.
- von Dyck and R-tables: $a, b \mapsto \alpha, \beta$ extends to group hom $\varphi\colon G \to \mathrm{Sym}_4$ defining a G-action on $\{1,2,3,4\}$.
- Construction and S-tables: G acts transitively, $U \leqslant G_1$, and $[G:U] \geqslant [G:G_1] = 4$.
- Labels represent cosets and every $g \in G$ lies in one of these, so $[G:U] \leqslant 4$, and $[G:U] = 4$, as claimed.

(See Handbook Thm 5.2)

Coset enumeration

Coset Enumeration

Let G be an fp group with $U \leqslant G$ of finite index. Using an *appropriate strategy*, the previous method will terminate and produce a permutation representation describing the G-action on right cosets of U. In particular, $[G:U]$ is determined.

Comments:

- Sometimes this is called the Todd-Coxeter 'Algorithm' (developed 1936).
- Different strategies exists (Felsch, or Haselgrove, Leech, Trotter).
- Tricky to implement efficiently; no strategy is optimal on all examples.
- **Problem:** How to know a priori whether $[G:U]$ is finite?

Further reading: Handbook, Sims, and the following survey:
Neubüser (1982): *An elementary introduction to coset table methods in computational group theory.*

Coset enumeration: runtime

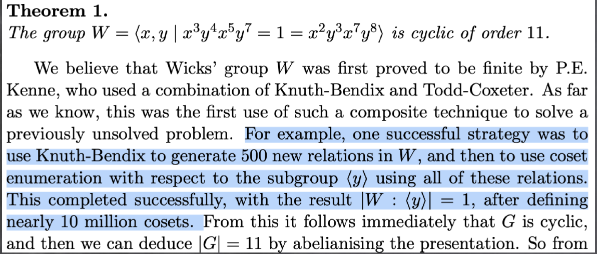
Proposition

The runtime for determining $[G:U]$ is not bounded by a function in $[G:U]$.

Proof. Suppose, for a contradiction, it is bounded by f. For a given fp group G, set $U=1$ and run the method $f(1)+1$ steps. If the process has not terminated, $G\neq 1$ is proved. If it has terminated, $|G|$ is finite. This constitutes a test for whether G is trivial, which is algorithmically undecidable (Rabin 1958). ↯

Johnson (1997): $\langle x,y \mid x^n y^{n+1}, x^{n+1}y^{n+2}\rangle$ is trival, but requires $\geqslant n$ cosets.

Revisiting Wicks group W:
(Havas, Holt, Kenne, Rees 1999)



Theorem 1.
The group $W=\langle x,y \mid x^3y^4x^5y^7 = 1 = x^2y^3x^7y^8\rangle$ *is cyclic of order* 11.

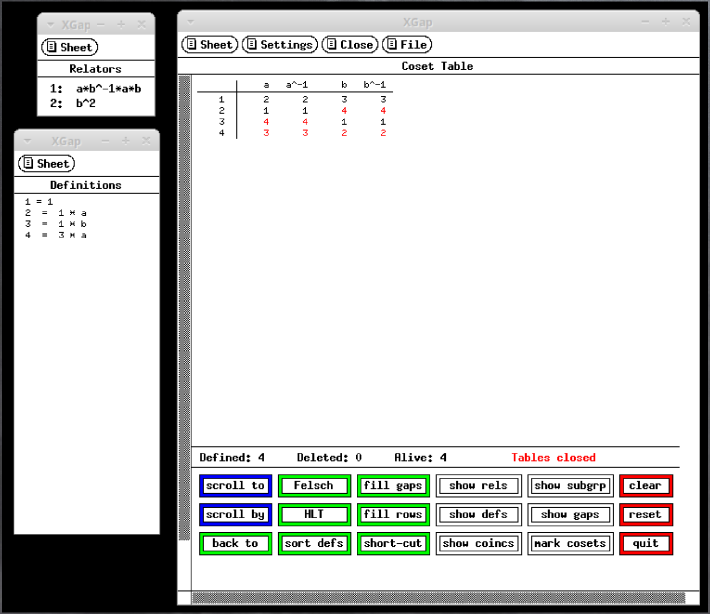
We believe that Wicks' group W was first proved to be finite by P.E. Kenne, who used a combination of Knuth-Bendix and Todd-Coxeter. As far as we know, this was the first use of such a composite technique to solve a previously unsolved problem. For example, one successful strategy was to use Knuth-Bendix to generate 500 new relations in W, and then to use coset enumeration with respect to the subgroup $\langle y\rangle$ using all of these relations. This completed successfully, with the result $|W:\langle y\rangle| = 1$, after defining nearly 10 million cosets. From this it follows immediately that G is cyclic, and then we can deduce $|G| = 11$ by abelianising the presentation. So from

GAP

```
gap> F := FreeGroup(["a","b"]);;
gap> G := F/ParseRelators(F,"a*a^b,b^2");;
gap> U := Subgroup(G,[G.1^2]);; # G.1 corresponds to first generator
gap> PrintArray(TransposedMat( CosetTable(G,U) ));
[ [  2,  2,  3,  3 ],
  [  1,  1,  4,  4 ],
  [  4,  4,  1,  1 ],
  [  3,  3,  2,  2 ] ]
# columns are sorted a, a^-1, b, b^-1
gap> List(CosetTable(G,U),PermList);
[ (1,2)(3,4), (1,2)(3,4), (1,3)(2,4), (1,3)(2,4) ]
gap>  Index(G,U);
4
```

```
gap> F := FreeGroup(["a","b"]);;
gap> W := F/ParseRelators(F,"a^3*b^4*a^5*b^7, a^2*b^3*a^7*b^8");;
gap> Size(W); # it is known that W has size 11, however:
#I  Coset table calculation failed -- trying with bigger table limit
#I  Coset table calculation failed -- trying with bigger table limit
...
Error, reached the pre-set memory limit [~16GB]
```

XGAP



Relators

1: a*b^-1*a*b
2: b^2

Definitions

```
1 = 1
2  =  1 * a
3  =  1 * b
4  =  3 * a
```

Coset Table

	a	a^-1	b	b^-1
1	2	2	3	3
2	1	1	4	4
3	4	4	1	1
4	3	3	2	2

Defined: 4   Deleted: 0   Alive: 4   Tables closed

▸ skip LIS

Low index subgroups & Reidemeister-Schreier

Let $G = \langle X \mid \mathcal{R} \rangle$ with $X = \{x_1, \ldots, x_m\}$ be an fp group.

Coset enumeration: Every subgroup $U \leqslant G$ of index n yields a *complete coset table* (incl. relator and subgroup tables) with n rows, such that $U = G_1$.

Conversely: A set of *complete* coset/relator tables with n symbols defines a transitive permutation rep. $G \to \mathrm{Sym}_n$, a subgroup $U = G_1$ with $[G : U] = n$, and a prefix-closed right transversal $T = \{t_1, \ldots, t_n\}$ of U in G with $t_1 = 1$.

Low-index subgroup algorithm: Compute subgroups $U \leqslant G$ of *small* index n by constructing complete coset/relator tables with n symbols.

Each such U is an fp group: (see the Handbook §2.5 & §5.3))

- For $g \in G$ let $\overline{g} \in T$ such that $Ug = U\overline{g}$. Then U is generated by the multiset of *Schreier generators* $S = \{s_{i,j} = t_i x_j (\overline{t_i x_j})^{-1} : x_j \in X, t_i \in T, s_{i,j} \neq 1\}$.
- $U \cong \langle S \mid \mathcal{T} \rangle$ for the *Reidemeister relators* $\mathcal{T} = \{\tau(trt^{-1}) : t \in T, r \in \mathcal{R}\}$; here τ is the *Reidemeister rewriting process* that produces words in $s_{i,j}$ . . .

  The idea is to rewrite $x_i x_j$ as $(t_1 x_i (\overline{t_1 x_i})^{-1}) \cdot (\overline{t_1 x_i} x_j (\overline{t_1 x_i x_j})^{-1}) \cdot \overline{t_1 x_i x_j}$, etc.

From table to group: Reidemeister-Schreier

One *complete* coset table for $G = \langle a, b \mid ab^{-1}ab, b^2\rangle$ on $n = 4$ points is:

C-table

	a	b	a^{-1}	b^{-1}
1	1,**2**	α,4	β^{-1},2	ε^{-1},4
2	β,1	1,**3**	1,1	δ^{-1},3
3	1,**4**	δ,2	γ^{-1},4	1,2
4	γ,3	ε,1	1,3	α^{-1},1

Transversal

$$\begin{aligned} t_1 &= 1 \\ t_2 &= a \\ t_3 &= ab \\ t_4 &= aba \end{aligned}$$

Schreier generators

$$\begin{aligned} \alpha &= ba^{-1}b^{-1}a^{-1} \\ \beta &= a^2 \\ \delta &= ab^2a^{-1} \\ \gamma &= aba^2b^{-1}a^{-1} \\ \varepsilon &= abab \end{aligned}$$

Schreier generators: compute $t_igt_{i^g}^{-1}$ for $g \in \{a, b\}$ and $i = 1, \ldots, 4$;
e.g., b maps $i = 3$ to 2, which yields $t_3bt_2^{-1} = abba^{-1}$. We obtain $\alpha, \ldots, \varepsilon$.

Amended C-table: in row i and column g, add label $t_igt_{i^g}^{-1}$.

Reidemeister relators: scan relators acting on cosets and multiply the labels;

e.g., $ab^{-1}ab$ on 2 scans as $2 \xrightarrow[\beta]{a} 1 \xrightarrow[\varepsilon^{-1}]{b^{-1}} 4 \xrightarrow[\gamma]{a} 3 \xrightarrow[\delta]{b} 2$ and yields relator $\beta\varepsilon^{-1}\gamma\delta$.

We obtain $G_1 \cong \langle \alpha, \ldots, \varepsilon \mid \alpha\varepsilon, \delta, \delta^{-1}\varepsilon, \alpha^{-1}, \gamma\beta\alpha, \beta\varepsilon^{-1}\gamma\delta\rangle \cong \langle \beta \mid \emptyset\rangle \cong C_\infty$.

Getting tables

Want: Subgroups of fp group G of index at most n.

Idea: Construct *complete* coset tables with at most n rows.

The **main idea** is to start coset enumeration for G with $U = \{1\}$.

If at some stage $n+1$ cosets $Uw_1, \ldots, Uw_{n+1}$ are constructed, then one can enforce a coincidence, say $Uw_i = Uw_j$, and replace U by $\langle U, w_i w_j^{-1} \rangle$.

A back-track search through the nodes of a suitable tree is used to create all complete tables with at most n rows (corresponding to subgroups of index at most n, *up to conjugacy*).

For details: Dietze-Schaps (1974), Sims (1974), Sims (1994), or the Handbook.

GAP

```
gap> F   := FreeGroup(["a","b"]);;
gap> G   := F/ParseRelators(F,"a*a^b,b^2");;
gap> lis := LowIndexSubgroups(G,10);; # takes 3ms
gap> List(lis, U->Index(G,U) );
[ 1, 2, 2, 3, 4, 5, 6, 7, 8, 9, 10, 2, 4, 6, 8, 10, 4, 6, 8, 10 ]
gap> GeneratorsOfGroup(lis[17]);      # our subgroup of index 4
[ a^-2 ]
gap> Index(G, Intersection(lis) );
5049
```

```
gap> F   := FreeGroup(["x","y","z"]);;
gap> G   := F/ParseRelators(F,"[x,[x,y]]=z,[y,[y,z]]=x,[z,[z,x]]=y");;
gap> lis := LowIndexSubgroups(G,10);; # takes 282s
gap> List(lis, U->Index(G,U) );       # intersections has index 120
[ 1, 6, 5, 10, 10 ]
```

Working with subgroups of fp groups in GAP:
coset tables, generators, or "quotient representations".

Hulpke (2001): Represent subgroup $U \leqslant G$ as (φ, V) where $\varphi\colon G \to Q$ is hom. into a group we can compute with, $V \leqslant Q$, and $U = \varphi^{-}(V)$ is the full preimage.

E.g., in coset enumeration, $\varphi\colon G \to Q \leqslant \mathrm{Sym}_n$ and $U = \varphi^{-}(Q_1)$.

Conclusion Lecture 1

Things we have discussed in the first lecture:

- free groups, group presentations, fp groups
- Dehn problems
- Tietze transformations
- von Dyck's Theorem
- Todd-Coxeter coset enumeration
- low-index subgroups
- Reidemeister-Schreier

Questions?

Second Lecture

▸ Go to Overview

Recap Lecture 1

Things we have discussed in the first lecture:

- Background: free groups, group presentations, fp groups
- Can't do: Dehn problems
- Can do: Tietze transformations, von Dyck's Theorem
- Can do (sometimes): Todd Coxeter coset enumeration
- Can do: Low-index subgroups, Reidemeister-Schreier

Today: more "*can do's*".

Quotient algorithms

Let $G = \langle X \mid \mathcal{R} \rangle$ be a finitely presented group. Recall that there are many questions that are algorithmically undecidable, and computing in G might be hard.

However: We might be able to use von Dyck's Theorem to find an epimorphism

$$\varphi\colon G \to H$$

onto a group H we *can* compute with: find an assignment of generator images $\kappa\colon X \to H$ that maps the defining relators $\mathcal{R}$ to 1_H and such that $H = \langle \kappa(X) \rangle$.

Useful: Since $H \cong G/\ker\varphi$, the structure of H tells us something about G.

This is the idea of **quotient algorithms**.

Often H is given as an fp group as well, so we first discuss a class of fp groups that we can compute with very well: **polycyclic groups**.

Polycyclic groups

A group H is **pc** (polycyclic) if it admits a polycyclic series

$$H = H_1 \trianglerighteq H_2 \trianglerighteq \ldots \trianglerighteq H_n \trianglerighteq H_{n+1} = 1,$$

that is, each quotient $H_i/H_{i+1} = \langle h_i H_{i+1} \rangle$ is cyclic, say of order $r_i \in \mathbb{N} \cup \{\infty\}$.

- $X = (h_1, \ldots, h_n)$ is a **pcgs** (polycyclic generating set);
- $R(X) = (r_1, \ldots, r_n)$ are the corresponding **relative orders**.
- Good for induction: each $H_i = \langle h_i, \ldots, h_n \rangle$ has pcgs $(h_i, \ldots, h_n)$.

Example: Let $D_{16} = \langle r, m \rangle$ with $r = (1, 2, \ldots, 8)$ and $m = (1, 3)(4, 8)(5, 7)$.
Examples of pcgs:

- $X = [m, r]$ with $R(X) = [2, 8]$: $G = \langle m, r \rangle > \langle r \rangle > 1$;
- $X = [m, r, r^4]$ with $R(X) = [2, 4, 2]$: $G = \langle m, r, r^4 \rangle > \langle r, r^4 \rangle > \langle r^4 \rangle > 1$;
- $X = [m, r, r^3, r^2]$ with $R(X) = [2, 1, 2, 4]$; note that $\langle r, r^3, r^2 \rangle = \langle r^3, r^2 \rangle$.

Polycyclic groups: normal forms

Let H be polycyclic with pcgs $X = (h_1, \ldots, h_n)$ and $R(X) = (r_1, \ldots, r_n)$.

Normal forms: Every $h \in H$ can uniquely be written as $h = h_1^{e_1} \ldots h_n^{e_n}$ with $e_i \in \{0, \ldots, r_i - 1\}$ if $r_i \neq \infty$, and $e_i \in \mathbb{Z}$ otherwise.

Proof. Induction on n: If $n = 1$, then H is cyclic ✓ Otherwise, there is a unique e_1 such that $hH_2 = h_1^{e_1} H_2$, and $h' = h_1^{-e_1} h \in H_2$. The induction hypothesis applied to H_2 implies that $h' = h_2^{e_2} \ldots h_n^{e_n}$, hence $h = h_1^{e_1} \ldots h_n^{e_n}$ as required.

Example: Let $H = \mathrm{Alt}_4$ and consider the pcgs

$$h_1 = (1,2,3), \quad h_2 = (1,2)(3,4), \quad h_3 = (1,3)(2,4)$$

with series $H = H_1 > \ldots > H_4 = 1$ and relative orders $[3, 2, 2]$. Let $h = (1, 2, 4)$:

- $hH_2 = h_1^2 H_2$, so $h' = h_1^{-2} h = (1,4)(2,3) \in H_2$.
- $h' H_3 = h_2 H_3$, so $h'' = h_2^{-1} h' = (1,3)(2,4) = h_3 \in H_3$.

In conclusion, $h = h_1^2 h' = h_1^2 h_2 h'' = h_1^2 h_2 h_3$.

Group to presentation

Let H be polycyclic with pcgs $(h_1,\ldots,h_n)$ and relative orders $(r_1,\ldots,r_n)$. These generators satisfy the following **power** and **conjugate relations**:

- If $r_i \neq \infty$, then $h_i^{r_i} \in H_{i+1}$ has a normal form, say $h_i^{r_i} = w_i(h_{i+1},\ldots,h_n)$.
- If $i<j$ then $H_j \leqslant H_{i+1} \trianglelefteq H_i$, so $h_j^{h_i^{\pm}} \in H_{i+1}$, say $h_j^{h_i^{\pm}} = w_{i,j}^{\pm}(h_{i+1},\ldots,h_n)$.

Now mimick this: Use these relations to define the fp group

$$\begin{array}{rcllcl} G & = & \langle g_1,\ldots,g_n \mid & \forall r_i \neq \infty & : & g_i^{r_i} = w_i(g_{i+1}\ldots,g_n), \\ & & & \forall i<j & : & g_j^{g_i^{\pm}} = w_{i,j}^{\pm}(g_{i+1},\ldots,g_n)\rangle. \end{array}$$

This group is polycyclic with pcgs $(g_1,\ldots,g_n)$; in particular $G \cong H$.

Proof. The relations imply that $G_i = \langle g_i,\ldots,g_n\rangle$ are the terms of a polycyclic series; also $(g_iG_{i+1})^{r_i} = G_{i+1}$ for $r_i \neq \infty$. As before, every $g \in G$ can be written as $g_1^{e_1}\ldots g_n^{e_n}$ with $0 \leqslant e_i < r_i$ if $r_i \neq \infty$ – *but is this unique?* Yes!

Von Dyck's Theorem shows that $g_1,\ldots,g_n \mapsto h_1,\ldots,h_n$ defines an epimorphism $\varphi\colon G \to H$. Since H has unique normal forms (wrt $r_1,\ldots,r_n$), φ is injective.

Example

Polycyclic group: $H = \langle (2,4,3), (1,3)(2,4)\rangle \cong \mathrm{Alt}_4$ has polycyclic series

$$H = H_1 > H_2 > H_3 > H_4 = 1$$

where $H_2 = \langle (1,3)(2,4),\ (1,2)(3,4)\rangle$ and $H_3 = \langle (1,2)(3,4)\rangle$.

Pcgs: Each H_i/H_{i+1} is cyclic, so there is $h_i \in H_i$ with $H_i/H_{i+1} = \langle h_i H_{i+1}\rangle$; eg

$$h_1 = (2,4,3), \quad h_2 = (1,3)(2,4), \quad h_3 = (1,2)(3,4),$$

and so (h_1, h_2, h_3) is a pcgs for H.

Presentation: These generators satisfy the following power/conjugate relations

- $h_1^3 = 1, \quad h_2^2 = 1, \quad h_3^2 = 1,$
- $h_2^{h_1} = (1,2)(3,4) = h_3, \quad h_3^{h_1} = (1,4)(2,3) = h_2 h_3, \quad h_3^{h_2} = h_3,$

so

$$H \cong \langle g_1, g_2, g_2 \mid g_1^3,\ \ g_2^2,\ \ g_3^2,\ \ g_2^{g_1} = g_3,\ \ g_3^{g_1} = g_2 g_3,\ \ g_3^{g_2} = g_3\rangle.$$

Polycyclic presentations

A **polycyclic presentation** (pcp) is a group presentation of the form

$$G = \langle g_1, \ldots, g_n \mid \begin{array}{lll} \forall i \in \mathcal{I} & : & g_i^{s_i} = w_i(g_{i+1} \ldots, g_n), \\ \forall i < j & : & g_j^{g_i^{\pm}} = w_{i,j}^{\pm}(g_{i+1}, \ldots, g_n) \rangle. \end{array}$$

where $s_1, \ldots, s_n \in \mathbb{N} \cup \{\infty\}$ with $s_i \neq \infty$ if and only if $i \in \mathcal{I} \subseteq \{1, \ldots, n\}$.

Observations:

- The group G is polycyclic with pcgs $(g_1, \ldots, g_n)$ and terms $G_i = \langle g_i, \ldots, g_n \rangle$.
- Every $g \in G$ can be written as $g = g_1^{e_1} \ldots g_n^{e_n}$ with $0 \leqslant e_i < s_i$ if $s_i \neq \infty$.
- The presentation is **consistent** if the latter form is unique.

Note: Consistency holds if and only if the *power exponents* s_i equal the relative orders r_i, that is, if and only if each $s_i = r_i \overset{\text{def}}{=} |G_i/G_{i+1}| = |g_i G_{i+1}|$.

Good news: If a pcp comes from a group with pcgs, then it is consistent.

Polycyclic presentations: notation

The pcp of the elementary abelian group $G = C_p^4$ is

$$G = \langle a, b, c, d \mid a^p, b^p, c^p, d^p, b^a = b, c^a = c, d^a = d, c^b = c, d^b = d, d^c = d \rangle.$$

Convention

In a pcp, one usually doesn't want to list trivial conjugate relations such as $b^a = b$; we write $\mathrm{Pc}\langle S \mid R \rangle$ to indicate that a presentation should be considered as a pcp, with the convention that missing conjugate relations are assumed to be trivial.

Much better: with this convention, $G = \mathrm{Pc}\langle a, b, c, d \mid a^p, b^p, c^p, d^p \rangle \cong C_p^4$.

Comments:

- Polycyclic presentations are useful to encode large (pc-)groups.[1]
- Elements are multiplied by concatenation and normalising (*collection*).
- Consistency of a presentation can be checked by evaluating a finite set of test words (*consistency checks*).

[1] A group is polycyclic if and only if it is solvable and every subgroup is finitely generated.

Collection algorithm

Let $G = \mathrm{Pc}\langle x_1, \ldots, x_n \mid \mathcal{R}\rangle$ with power exponents $S = [s_1, \ldots, s_n]$.

$w = x_{i_1}^{e_1} \cdots x_{i_r}^{e_r}$ is **collected** (wrt S) if $i_1 < \ldots < i_r$ and $0 \leqslant e_i < s_i$ for $s_i \neq \infty$; if w is not collected, then $i_j > i_{j+1}$ for some j, or $e_j \notin [0, s_j)$ for some $s_j < \infty$.

A **collection algorithm** follows specific rules to bring w into collected form (by using the relations of G), e.g.:

- **Collection to the left**: move all occurrences of x_1 to the beginning of the word; next, move all occurrences of x_2 left until adjacent to the x_1's, etc.
- **Collection from the right**: the minimal non-normal subword nearest to the end of a word is selected.
- **Collection from the left**: the minimal non-normal subword nearest to the beginning of a word is selected.

Leedham-Green & Soicher (1990, 1998): experiments with collection strategies for p-groups "*collection from the left is a very good strategy*"; symbolic collection ("*deep thought*");
Newman & Niemeyer 2015: complexity in finite solvable groups. Open: *Is poly-time possible?*

Example: collection

Consider the pc-group

$$\begin{aligned} D_{16} \cong \mathrm{Pc}\langle x_1, x_2, x_3, x_4 \quad \mid \quad & x_1^2 = 1,\ \ x_2^2 = x_3x_4,\ \ x_3^2 = x_4,\ \ x_4^2 = 1, \\ & x_2^{x_1} = x_2x_3,\ \ x_3^{x_1} = x_3x_4\rangle. \end{aligned}$$

Aim: collect the word $x_3x_2x_1$.
Since power exponents are all “2”, we only use generator indices:

”to the left”

$$\begin{aligned} 3\underline{21} &= \underline{31}23 \\ &= 13\underline{42}3 \\ &= 1\underline{32}43 \\ &= 123\underline{43} \\ &= 12\underline{33}4 \\ &= 12\underline{44} \\ &= 12 \end{aligned}$$

“from the right”

$$\begin{aligned} 3\underline{21} &= \underline{31}23 \\ &= 13\underline{42}3 \\ &= 132\underline{43} \\ &= 1\underline{32}34 \\ &= 12\underline{33}4 \\ &= 12\underline{44} \\ &= 12 \end{aligned}$$

”from the left”

$$\begin{aligned} \underline{32}1 &= 2\underline{31} \\ &= \underline{21}34 \\ &= 12\underline{33}4 \\ &= 12\underline{44} \\ &= 12 \end{aligned}$$

GAP

Here's one way to define polycyclic groups (via pcp) in GAP:

Consider the pc group G with generators a, b, c, d, e, f and relators/relations

$$a^2, b^2, c^3, d^3, e^5, f^5,\ c^a = c^2,\ d^a = d^2,\ e^c = ef^3,\ f^a = e^4f^4,\ f^c = e^4f^3.$$

```
gap> coll:=FromTheLeftCollector(6);;
gap> ord:=[2,2,3,3,5,5];;
gap> for i in [1..6] do SetRelativeOrder(coll,i,ord[i]); od;;
gap> SetConjugate(coll,3,1,[3,2]);
gap> SetConjugate(coll,4,1,[4,2]);
gap> SetConjugate(coll,5,3,[5,1,6,3]);
gap> SetConjugate(coll,6,1,[5,4,6,4]);
gap> SetConjugate(coll,6,3,[5,4,6,3]);
gap> G:=PcpGroupByCollector(coll);
Pcp-group with orders [ 2, 2, 3, 3, 5, 5 ]
gap> AssignGeneratorVariables(G);;
#I  Assigned the global variables [ g1, g2, g3, g4, g5, g6 ]
gap> g6*g5*g4*g3*g2*g1;
g1*g2*g3^2*g4^2*g5^4*g6^4
gap> Exponents(last);
[ 1, 1, 2, 2, 4, 4 ]
```

Consistent pc presentations

Recall: A pcp with power exponents S is consistent if and only if every group element has a unique normal form with respect to S.

Example 5

The group $G = \mathrm{Pc}\langle a, b, c \mid a^3 = c,\ b^2 = c,\ c^5 = 1,\ b^a = bc\rangle$ has pcgs $X = [a, b, c]$ and power exponents $S = [3, 2, 5]$. We show $R(X) = [3, 2, 1]$, so $|G| = 6$:

- First, note that $b^{10} = c^5 = 1$, so $|b| \mid 10$.
- Second, $b^a = bc = b^3$ so $b^{27} = b^{(a^3)} = b^c = b^{(b^2)} = b$, and thus $|b| \mid 26$.
- This implies that $5 \nmid |b|$, and forces $c = 1$ in G.

Thus, $a^0b^0c^0 = 1 = a^0b^0c^1$ are two distinct normal forms wrt power exponents.

How to check consistency? $\rightsquigarrow$ use **collection** and **consistency checks**!

Consistency checks

$\mathrm{Pc}\langle x_1,\ldots,x_n \mid \mathcal{R}\rangle$ with power exps $[s_1,\ldots,s_n]$, each $s_i<\infty$, is consistent if and only if the normal forms of the following pairs of words coincide

$$\begin{array}{ll} x_k(x_jx_i) \text{ and } (x_kx_j)x_i & \text{for } 1\leqslant i<j<k\leqslant n,\\ (x_j^{s_j})x_i \text{ and } x_j^{s_j-1}(x_jx_i) & \text{for } 1\leqslant i<j\leqslant n,\\ x_j(x_i^{s_i}) \text{ and } (x_jx_i)x_i^{s_i-1} & \text{for } 1\leqslant i<j\leqslant n,\\ x_j(x_j^{s_j}) \text{ and } (x_j^{s_j})x_j & \text{for } 1\leqslant j\leqslant n, \end{array}$$

where the subwords in brackets are to be collected first; see Prop. 8.3 in Sims.
(*If $s_i=\infty$ or $s_j=\infty$, also consider $x_jx_i^{-1}x_i,\ x_j^{-1}x_jx_i,\ x_j^{-1}x_jx_i^{-1}$ for $i<j$.*)

Example 6

If $G=\mathrm{Pc}\langle a,b,c \mid a^3=c,\ b^2=c,\ c^5=1,\ b^a=bc\rangle$, then

$$(b^2)a=ca=ac \quad\text{and}\quad b(ba)=babc=ab^2c^2=ac^3.$$

Since ac and ac^3 are both normal forms, the presentation is *not* consistent.
Indeed, we deduce from $ac=ac^3$ that $c=1$ in G.

Now let’s consider quotient algorithms . . .

Computing the largest abelian quotient

It *can be difficult* to compute with an fp group G, but it's *easy* to compute with abelian groups! So let's find abelian quotients of G to work with.

Task: For an fp group G, compute its largest abelian quotient[2] G/G'.

Abelianisation

If $G = \langle X \mid \mathcal{R} \rangle$, then G/G' is isomorphic to $A = \langle X \mid \mathcal{R} \cup \{[x, y] : x, y \in X\} \rangle$.

Proof. $2\times$ von Dyck: first, $x \mapsto x$ defines an epi. $G \to A$ with G' in its kernel, so $G/G' \to A$ with $xG' \mapsto x$. Second, $x \mapsto xG'$ extends to an epi. $A \to G/G'$.

Fundemental Theorem of finitely generated abelian groups

If G is finitely generated, there exist *unique* integers n, s and $2 \leqslant d_1, \ldots, d_n$ with

$$G/G' \cong C_{d_1} \times \ldots \times C_{d_n} \times C_\infty^s$$

and $d_1 \mid \ldots \mid d_n$; the **abelian invariants** of G are $(d_1, \ldots, d_n; s)$.

[2] Commutator subgroup: $G' = \langle [x, y] : x, y \in G \rangle$ where each $[x, y] = x^{-1}y^{-1}xy$.

Computing the largest abelian quotient

To compute G/G' for $G = \langle x_1, \ldots, x_n \mid r_1, \ldots, r_m \rangle$, do the following:

1. *abelianise* each relator $r_j = x_{i_1}^{f_1} \ldots x_{i_\ell}^{f_\ell}$: i.e. write $r_j = x_1^{e_{j,1}} \ldots x_n^{e_{j,n}}$ in G/G'
2. write these exponents in an $m \times n$ matrix M with rows $(e_{j,1}, \ldots, e_{j,n})$,
3. observe that $G/G' \cong \langle X \mid \mathcal{R} \cup \{[x,y] : x,y \in X\} \rangle \cong \mathbb{Z}^n/\mathrm{Row}(M)$,
4. compute[3] $\mathrm{SNF}(M) = \mathrm{diag}(d_1, \ldots, d_k, 0, \ldots, 0)$ with each $1 \leqslant d_i \mid d_{i+1}$,
5. now $G/G' \cong \mathbb{Z}^n/\mathrm{Row}(M) \cong \mathbb{Z}^n/\mathrm{Row}(\mathrm{SNF}(M)) \cong C_{d_1} \times \ldots \times C_{d_k} \times C_\infty^{n-k}$.

Example

Let $G = \langle a,b,c,d \mid a^2b^2ca^2d^8,\ (a^{-2}b)^2cd^{-6}cd^{-2}b^2,\ b^{-1}a^4d^2c^d,\ ab^2ac^3dab^3cda \rangle$.
Compute the abelianised exponent matrix and its SNF:

$$M = \begin{pmatrix} 4 & 2 & 1 & 8 \\ -4 & 4 & 2 & -8 \\ 4 & -1 & 1 & 2 \\ 4 & 5 & 4 & 2 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \mathrm{SNF}(M).$$

Thus, $G/G' \cong \mathbb{Z}^4/\mathrm{Row}(\mathrm{SNF}(M)) \cong C_3 \times C_{12} \times C_\infty$; in particular, $|G| = \infty$.

[3] Smith-Normal-Form: $\mathrm{SNF}(M) = RMC$ for invertible integral mats R, C; arrange $m \geqslant n$. See also Havas & Sterling'79: *Integer matrices and abelian groups*.

Computing the largest abelian quotient

Example

Consider the Wicks group $W = \langle a, b \mid a^3b^4a^5b^7, a^2b^3a^7b^8 \rangle$.
The abelianised exponent matrix is

$$M = \left(\begin{smallmatrix} 8 & 11 \\ 9 & 11 \end{smallmatrix}\right) \rightsquigarrow \mathsf{SNF}(M) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 11 \end{smallmatrix}\right)$$

and therefore $W/W' \cong C_{11}$.

An immediate consequence of this method is the following:

If $G = \langle X \mid \mathcal{R} \rangle$ satisfies $|X| > |\mathcal{R}|$, then the SNF of the abelianised exponent matrix has a 0 on its diagonal, and then $|G| = \infty$.

E.g., the group $G = \langle a, b, c \mid ab^2c^{-2}a,\ b^3c^{-3}ac^9a \rangle$ is infinite.

p-quotients

One can also compute well with finite p-groups (via their pcp!); this leads to …

Task: For an fp group G, compute its largest finite p-group quotient.

But, wait … is there a largest finite p-group quotient? Not nesessarily, e.g.,

$$D_\infty = \langle a, b \mid a^2, b^a = b^{-1} \rangle$$

has every dihedral group $D_{2^{n+1}} \cong D_\infty / \langle b^{(2^n)} \rangle$ as a quotient.

Finite p-groups are nilpotent, so the following seems useful:

Lower Central Series

A group P is nilpotent if and only if its lower central series

$$P = \gamma_1(P) \geqslant \gamma_2(P) \geqslant \dots$$

with $\gamma_{i+1}(P) = [P, \gamma_i(P)]$ terminates in the trivial subgroup.

Thus, if G/N is a finite p-group quotient of G, then $\gamma_j(G) \leqslant N$ for some j.

Special central series are useful

Lower exponent-p series

The lower exponent-p series of a group G is

$$G = P_0(G) > P_1(G) > \dots$$

where each $P_{i+1}(G) = [G, P_i(G)]P_i(G)^{[p]}$; here we define $H^{[p]} = \langle h^p : h \in H\rangle$.
The p-class of G is c if $P_{c-1}(G) > P_c(G) = 1$.

Useful properties:

- each $P_i(G)$ is characteristic in G
- each $P_i(G)/P_{i+1}(G)$ is G-central elementary abelian
- if θ is a homomorphism from G, then $\theta(P_i(G)) = P_i(\theta(G))$
- G/N has p-class c if and only if c is minimal with $P_c(G) \leqslant N$
- for a p-group G: $P_1(G) = \Phi(G)$ is the Frattini subgroup, and

$$G/P_1(G) \cong C_p^d \quad \text{where } d = \text{rank}(G).$$

Example: lower exponent-p series

Example 7

Consider

$$G = D_{16} = \mathrm{Pc}\langle a_1, a_2, a_3, a_4 \mid a_1^2 = 1,\ a_2^2 = a_3a_4,\ a_3^2 = a_4,\ a_4^2 = 1,\ [a_2, a_1] = a_3,\ [a_3, a_1] = a_4\rangle.$$

Note: $[a_2, a_1] = a_3$ is equivalent to $a_2^{a_1} = a_2a_3$, etc.

Here we can read off:

- $P_0(G) = G$
- $P_1(G) = [G, G]G^{[2]} = \langle a_3, a_4\rangle$
- $P_2(G) = [G, P_1(G)]P_1(G)^{[2]} = \langle a_4\rangle$
- $P_3(G) = [G, P_2(G)]P_2(G)^{[2]} = 1$

So G has 2-class 3.

Special pcps are useful

A weighted pcp (wpcp) of a d-generator group of order p^n is $\mathrm{Pc}\langle x_1,\ldots,x_n \mid \mathcal{R}\rangle$ such that $\{x_1,\ldots,x_d\}$ is a minimal generating set G and the relations are

$$x_j^p = x_{j+1}^{\alpha_{j,j+1}} \ldots x_n^{\alpha_{j,n}} \quad \text{and} \quad [x_j,x_i] = x_{j+1}^{\beta_{i,j,j+1}} \ldots x_n^{\beta_{i,j,n}} \quad (i<j),$$

such that each $0 \leqslant \alpha_{u,v}, \beta_{u,v,w} < p$. For each $k>d$ there is one relation with right side x_k: the definition of x_k. The associated weight function is

$$\omega(x_k) = \begin{cases} 1 & (1 \leqslant k \leqslant d) \\ \omega(x_i)+1 & (x_k = x_i^p \text{ def.}) \\ \omega(x_j)+\omega(x_i) & (x_k = [x_j,x_i] \text{ def.}) \end{cases}$$

E.g.: $\{x_1,x_2\}$ is a minimal generating set and x_3,x_4,x_5 have weight $2,2,3$ in

$$G = \mathrm{Pc}\langle\ x_1,\ldots,x_5 \ \mid\ x_1^2 = x_4,\ x_2^2 = x_3,\ x_3^2 = x_5,\ x_4^2 = x_5,\ x_5^2 = 1$$
$$[x_2,x_1] = x_3,\ [x_3,x_1] = x_5\ \rangle.$$

ANUPQ: Computing the quotient $G/P_c(G)$

The following is mainly due to MacDonald'74, Newman'76, Havas-Newman'80; Newman-O'Brien'96; Newman-Nickel-Niemeyer'98; known as **ANUPQ**.

p-quotient algorithm

Input: an fp group $G = F/R = \langle x_1, \ldots, x_n \mid \mathcal{R} \rangle$, prime p, p-class c
Output: epimorphism $G \to G/P_c(G)$ and wpcp of $G/P_c(G)$.

Top-level outline:

- compute wpcp of $G/P_1(G)$ and epimorphism $G \to G/P_1(G)$, then iterate:
- given wpcp of $G/P_k(G)$ and epimorphism $G \to G/P_k(G)$, compute wpcp of $G/P_{k+1}(G)$ and epimorphism $G \to G/P_{k+1}(G)$;
- stop when $G/P_c(G)$ and epimorphism $G \to G/P_c(G)$ are computed.

First: get wpcp for $G/P_1(G)$

Input: an fp-group $G = \langle x_1, \ldots, x_n \mid \mathcal{R} \rangle$, prime p
Output: a wpcp of $G/P_1(G)$ and epimorphism $\theta\colon G \to G/P_1(G)$

Example: For $\langle x_1, \ldots, x_6 \mid x_6^{10},\ x_1x_2x_3,\ x_2x_3x_4, \ldots, x_4x_5x_6,\ x_5x_6x_1,\ x_1x_6x_2 \rangle$ and $p = 2$, write coefficients of abelianised and mod-2 reduced relators as rows of matrix, use row-echelonisation, and determine that solution space has dimension 2:

$$\begin{pmatrix} 1&1&1&0&0&0\\ 0&1&1&1&0&0\\ 0&0&1&1&1&0\\ 0&0&0&1&1&1\\ 1&0&0&0&1&1\\ 1&1&0&0&0&1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1&0&0&0&1&1\\ 0&1&0&0&1&0\\ 0&0&1&0&0&1\\ 0&0&0&1&1&1\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix};$$

Modulo $P_1(G)$, we have $x_1 = x_5x_6$, $x_2 = x_5$, $x_3 = x_6$, $x_4 = x_5x_6$, so x_5, x_6 map to a generating set of $G/P_1(G)$. A wpcp for $G/P_1(G)$ is

$$G/P_1(G) = \mathrm{Pc}\langle a_1, a_2 \mid a_1^2 = a_2^2 = 1 \rangle \cong C_2^2,$$

and we define $\theta\colon G \to G/P_1(G)$ via $x_5 \mapsto a_1$ and $x_6 \mapsto a_2$.
This determines $\theta(x_1) = a_1a_2$, $\theta(x_2) = a_1$, $\theta(x_3) = a_2$, and $\theta(x_4) = a_1a_2$.

Second: lift wpcp from $G/P_k(G)$ to $G/P_{k+1}(G)$

Given: epimorphism $\theta\colon G \to G/P_k(G)$ onto d-generator p-group with wpcp

Want: wpcp of $G/P_{k+1}(G)$ and epimorphism $G \to G/P_{k+1}(G)$

In the following:

- $H = G/P_k(G)$ and $K = G/P_{k+1}(G)$ and $Z = P_k(G)/P_{k+1}(G)$
- note that Z is elementary abelian, K-central, and $K/Z \cong H$

Approach: Construct a *covering* H^* of H such that *every* d-generator p-group L with $L/M \cong H$ and $M \leqslant L$ central elementary abelian, is a quotient of H^*.

Thus, the next steps are:

- define p-cover H^* and determine a pcp of H^*;
- make this presentation consistent;
- construct K as quotient of H^* by enforcing defining relations of G.

$$\begin{tikzcd}
 & H^* \arrow[d] \\
G \arrow[r, "\tilde{\theta}"] \arrow[dr, "\theta"'] & K \arrow[d] \\
 & H
\end{tikzcd}$$

p-covering group: definition

Theorem 8: p-covering group

Let H be a d-generator p-group; there is a d-generator p-group H^* with:

- $H^*/M \cong H$ for some central elementary abelian $M \trianglelefteq H^*$;
- if L is a d-generator p-group with $L/Y \cong H$ for some central elementary abelian $Y \leqslant L$, then L is a quotient of H^*.

The group H^* is unique up to isomorphism.

Proof. Let $H = F/S$ with F free of rank d; let $\theta\colon F \to H$ with kernel S.

Set $H^* = F/S^*$ with $S^* = [S, F]S^{[p]}$.

Then S/S^* is elementary abelian and H^* is a finite d-generator p-group.

Let L be as in the theorem, and let $\psi\colon L \to H$ with central el.-ab. kernel Y.

Since F is free, θ factors through L, say $\theta\colon F \xrightarrow{\varphi} L \xrightarrow{\psi} H$ with $\varphi(S) \leqslant \ker\psi = Y$.

This implies $\varphi(S^*) = 1$, and φ induces epimorphism $H^* = F/S^* \to L$.

If H^* and $\tilde{H}^*$ are both covers, then each is an image of the other, so $H^* \cong \tilde{H}^*$.

p-covering group: presentation

Given: a wpcp $\mathrm{Pc}\langle a_1,\ldots,a_m \mid \mathcal{S}\rangle$ for $H = G/P_k(G) \cong F/S$
and epimorphism $\theta\colon G \to H$ with $\theta(x_i) = a_i$ for $i = 1,\ldots,d$

Want: a wpcp for $H^* \cong F/S^*$ where $S^* = [S,F]S^p$

Recall: each of $a_{d+1},\ldots,a_m$ occurs as right hand side of one relation in $\mathcal{S}$;
write $\mathcal{S} = \mathcal{S}_{\text{def}} \cup \mathcal{S}_{\text{nondef}}$ with $\mathcal{S}_{\text{nondef}} = \{s_1,\ldots,s_q\}$.

Theorem 9: presentation of cover

Using the previous notation, $H^* = \mathrm{Pc}\langle a_1,\ldots,a_m, b_1,\ldots,b_q \mid \mathcal{S}^*\rangle$, where

$$\mathcal{S}^* = \mathcal{S}_{\text{def}} \cup \{s_1b_1,\ldots,s_qb_q\} \cup \{b_1^p,\ldots,b_q^p\}.$$

Note: $H^*/M \cong H$ where $M = \langle b_1,\ldots,b_q\rangle \trianglelefteq H^*$ is central elementary abelian.

(see Newman, Nickel, Niemeyer: "*Descriptions of groups of prime-power order*", 1998)

In practice: fewer new generators are introduced.

p-covering group: example

Example: If $H = \mathrm{Pc}\langle a_1, a_2 \mid a_1^2 = a_2^2 = 1\rangle \cong C_2 \times C_2$, then

$$H^* = \mathrm{Pc}\langle a_1, a_2, b_1, b_2, b_3 \mid a_1^2 = b_1,\ a_2^2 = b_2,\ [a_2, a_1] = b_3,\ b_1^2 = b_2^2 = b_3^2 = 1\rangle;$$

indeed, $H^* \cong C_4 \ltimes (C_4 \times C_2)$, thus we have found a consistent wpcp!

Example: If $H = \mathrm{Pc}\langle a_1, a_2, a_3 \mid a_1^2 = a_3^2 = 1, a_2^2 = a_3, [a_2, a_1] = a_3\rangle \cong D_8$, then

$$H^* = \mathrm{Pc}\langle a_1, a_2, a_3, b_1, \ldots, b_5 \mid \mathcal{T} \cup \{b_1^2, \ldots, b_5^2\}\ \rangle \quad \text{with}$$

$$\mathcal{T} = \{a_1^2 = b_1, a_2^2 = a_3 b_2, a_3^2 = b_3, [a_2, a_1] = a_3, [a_3, a_1] = b_4, [a_3, a_2] = b_5\};$$

this pcp has power exponents $[2, 2, 2, 2, 2, 2, 2, 2]$.

However, $H^* \cong C_4 \ltimes (C_8 \times C_2)$, so this presentation is **not consistent**!

Next step: make the presentation of H^* consistent (*not here!*).

Construct K from cover H^* of H

So what have we got so far...

- fp-group $G = F/R = \langle x_1, \ldots, x_n \mid \mathcal{R} \rangle$ and a fixed prime p
- consistent wpcp of $H = G/P_k(G) = \mathrm{Pc}\langle a_1, \ldots, a_m \mid \mathcal{S} \rangle$
- epimorphism $\theta\colon G \to H$ with $\theta(x_i) = a_i$ for $i = 1, \ldots, d$
- consistent wpcp of cover $H^* = \mathrm{Pc}\langle a_1, \ldots, a_m, b_1, \ldots, b_q \mid \mathcal{S}^* \rangle$; note that $H^*/M \cong H$ where $M = \langle b_1, \ldots, b_q \rangle$

Want:

- consistent wpcp of $K = G/P_{k+1}(G)$ and epimorphism $G \to G/P_{k+1}(G)$

Know:

- $K/Z \cong H$ where $Z = P_k(G)/P_{k+1}(G)$ is elementary abelian, central
- K is quotient of H^*

Idea:

- construct K as quotient of H^*: add relations enforced by G to $\mathcal{S}^*$

Construct K from cover H^* of H

Recall:

- $G = F/R = \langle x_1, \ldots, x_n \mid \mathcal{R}\rangle$ and $H = G/P_k(G)$.
- The epimorphism $\theta\colon G \to H$ maps $x_1, \ldots, x_d$ to $a_1, \ldots, a_d$.
- Cover H^* maps onto $H \cong H^*/M$ and $K = G/P_{k+1}(G)$.

Enforcing relations of G:

$$\begin{tikzcd}
F \arrow[r, "\hat{\theta}"] \arrow[d] & H^* \arrow[d] \\
G \arrow[r, dashed] \arrow[rd, "\theta"'] & K \arrow[d] \\
& H
\end{tikzcd}$$

- lift θ to $\hat{\theta}\colon F \to H^*$ with $\hat{\theta}(x_i) = a_i$ for $i = 1, \ldots, d$

  (Subtlety: for $j > d$, add new temporary generators c_j of M such that $\hat{\theta}(x_j) = \theta(x_j)c_j$; see Handbook p. 361, Havas & Newman'80)
- if $r \in \mathcal{R}$, then $\hat{\theta}(r) \in M$; let $L = \langle \hat{\theta}(r) : r \in \mathcal{R}\rangle \leqslant M$
- by von Dyck, the assignment $x_i \mapsto a_i$ yields epimorphisms $G \to H^*/L$ and $H^*/L \to K$.

In fact, $K = H^*/L$ since K is largest p-class $k+1$ quotient of G.

Big example: p-quotient algorithm in action

Let $G = \langle x, y \mid [[y,x],x] = x^2,\ (xyx)^4,\ x^4,\ y^4,\ (yx)^3y = x\rangle$ and $p = 2$. ▸ no!

First round:

- compute $G/P_1(G)$ using abelianisation and row-echelonisation:

  obtain $H = G/P_1(G) \cong \mathrm{Pc}\langle a_1, a_2 \mid a_1^2 = a_2^2 = 1\rangle$
  and epimorphism $\theta\colon G \to H$, which is defined by $(x,y) \to (a_1, a_2)$.
- construct covering of H by adding new generators and tails:

  $H^* = \mathrm{Pc}\langle a_1, \ldots, a_5 \mid a_1^2 = a_3, a_2^2 = a_4, [a_2, a_1] = a_5,\ a_3^2 = a_4^2 = a_5^2 = 1\rangle$
- the consistency algorithm shows that this presentation is consistent
- evaluate relations of G in H^*:
  - $1 = [[a_2, a_1], a_1] = \hat\theta([[y,x],x]) = \hat\theta(x^2) = a_1^2 = a_3$ forces $a_3 = 1$
  - $(xyx)^4, x^4, y^4$ impose no conditions
  - $a_1a_3 = (a_2a_1)^3a_2 = \hat\theta((yx)^3y) = \hat\theta(x) = a_1$ also forces $a_3 = 1$
- construct $G/P_2(G)$ as $H^*/\langle a_3\rangle$; after renaming a_5:

  $$G/P_2(G) \cong \mathrm{Pc}\langle a_1, \ldots, a_4 \mid a_1^2 = 1, a_2^2 = a_4, [a_2, a_1] = a_3,\ a_3^2 = a_4^2 = 1\rangle$$

  and epimorphism $G \to G/P_2(G)$ defined by $(x,y) \to (a_1, a_2)$.

Big example: p-quotient algorithm in action

$G/P_2(G) = \mathrm{Pc}\langle a_1, \ldots, a_4 \mid a_1^2 = 1, a_2^2 = a_4, [a_2, a_1] = a_3,\ a_3^2 = a_4^2 = 1\rangle$

Second round:

- construct covering of $H = G/P_2(G)$ by adding new generators and tails:

  $$\begin{aligned} H^* = \mathrm{Pc}\langle a_1, \ldots, a_{12} \mid\ & a_1^2 = a_{12}, a_2^2 = a_4, a_3^2 = a_{11}, a_4^2 = a_{10},\\ & [a_2, a_1] = a_3, [a_3, a_1] = a_5, [a_3, a_2] = a_6, [a_4, a_1] = a_7,\\ & [a_4, a_2] = a_8, [a_4, a_3] = a_9,\ a_5^2 = \ldots = a_{12}^2 = 1\rangle \end{aligned}$$

- the consistency algorithm shows only the following inconsistencies:
  - $a_2(a_2a_2) = a_2a_4$ and $(a_2a_2)a_2 = a_4a_2 = a_2a_4a_8$ $\Longrightarrow a_8 = 1$
  - $a_2(a_1a_1) = a_2a_{12}$ and $(a_2a_1)a_1 = a_1a_2a_3a_1 = \ldots = a_2a_5a_{11}a_{12}$ $\Longrightarrow a_5a_{11} = 1$
  - $a_2(a_2a_1) = a_1a_2^2a_3^2a_6 = a_1a_4a_6a_{11}$ and $(a_2a_2)a_1 = a_1a_4a_7$ $\Longrightarrow a_6a_7a_{11} = 1$
  - $a_3(a_2a_2) = a_3a_4$ and $(a_3a_2)a_2 = a_2a_3a_6a_2 = a_2^2a_3a_6^2 = a_3a_4a_9$ $\Longrightarrow a_9 = 1$

- removing redundant gens (and renaming), we obtain the consistent wpcp

  $$\begin{aligned} H^* = \mathrm{Pc}\langle a_1, \ldots, a_8 \mid\ & a_1^2 = a_8, a_2^2 = a_4, a_3^2 = a_7, a_4^2 = a_6,\ a_5^2 = \ldots = a_8^2 = 1\\ & [a_2, a_1] = a_3, [a_3, a_1] = a_7, [a_3, a_2] = a_5a_7, [a_4, a_1] = a_5\rangle \end{aligned}$$

Big example: p-quotient algorithm in action

Still second round:

- $G = \langle x, y \mid [[y,x],x] = x^2, (xyx)^4, x^4, y^4, (yx)^3y = x\rangle$ and $p = 2$;
- epimorphism $\theta\colon G \to H$ onto $H = G/P_2(H)$ defined by $(x,y) \to (a_1,a_2)$
- $H^* = \mathrm{Pc}\langle a_1,\ldots,a_8 \mid a_1^2 = a_8, a_2^2 = a_4, a_3^2 = a_7, a_4^2 = a_6,\ a_5^2 = \ldots = a_8^2 = 1$
  $[a_2,a_1] = a_3, [a_3,a_1] = a_7, [a_3,a_2] = a_5a_7, [a_4,a_1] = a_5\rangle$

Evaluate relations of G in H^*:

- $a_7 = [[a_2,a_1],a_1] = \hat\theta([[y,x],x]) = \hat\theta(x^2) = a_1^2 = a_8$ forces $a_7 = a_8$
- $(xyx)^4$ forces $a_6 = 1$; x^4 and y^4 impose no condition
- $\hat\theta((yx)^3y) = \hat\theta(x)$ forces $a_7a_8 = 1$

Now construct $G/P_3(G)$ as $H^*/\langle a_7a_8, a_6\rangle$; after renaming:

$$G/P_3(G) = \mathrm{Pc}\langle a_1,\ldots,a_6 \mid a_1^2 = a_6, a_2^2 = a_4, a_3^2 = a_6, a_4^2 = 1,\ a_5^2 = a_6^2 = 1,$$
$$[a_2,a_1] = a_3, [a_3,a_1] = a_6, [a_3,a_2] = a_5a_6, [a_4,a_1] = a_5\rangle$$

and the epimorphism $G \to G/P_3(G)$ is defined by $(x,y) \to (a_1,a_2)$.

Big example: p-quotient algorithm in action

In conclusion:

We started with

$$G = \langle x, y \mid [[y,x],x] = x^2, (xyx)^4, x^4, y^4, (yx)^3y = x\rangle$$

and computed $G/P_3(G)$ as

$$\begin{aligned}\mathrm{Pc}\langle a_1,\ldots,a_6 \mid\ & a_1^2 = a_6,\ a_2^2 = a_4,\ a_3^2 = a_6,\ a_4^2 = a_5^2 = a_6^2 = 1,\\ & [a_2,a_1] = a_3, [a_3,a_1] = a_6, [a_3,a_2] = a_5a_6, [a_4,a_1] = a_5\rangle\end{aligned}$$

with epimorphism $G \to G/P_3(G)$ defined by $(x,y) \to (a_1,a_2)$.

One can check that $|G| = |G/P_3(G)| = 2^6$, hence $G \cong G/P_3(G)$.

In particular, we have found a consistent wpcp for G.

Big example: GAP

```
gap> F    := FreeGroup("x","y");;
gap> G    := F/ParseRelators(F,"[[y,x],x]=x^2,(x*y*x)^4,x^4,y^4,(y*x)^3*y=x");;
gap> epi1 := EpimorphismPGroup(G,2,1);; #2-class 1 quotient
gap> StructureDescription(Image(epi1));
"C2 x C2"
gap> epi2 := EpimorphismPGroup(G,2,2);; #2-class 2 quotient
gap> StructureDescription(Image(epi2));
"(C4 x C2) : C2"
gap> epi3 := EpimorphismPGroup(G,2,3);; #2-class 3 quotient
gap> StructureDescription(Image(epi3));
"(C2 x Q8) : C4"
gap> Size(Image(epi3))=Size(G);
true
gap> #now compute the 2-cover of G/P_1(G)
gap> LoadPackage("anupq");
true
gap> H   := Image(epi1);;
gap> Hs  := PqPCover(H);;
gap> StructureDescription(Hs);
"(C4 x C2) : C4"
```

Application: Burnside groups

Burnside Problems

- Generalised Burnside Problem (GBP), 1902:
  Is every finitely generated torsion group finite?
- Burnside Problem (BP), 1902:
  Let $B(d,n)$ be the largest d-generator group with $g^n = 1$ for all $g \in G$.
  Is this group finite? If so, what is its order?
- Restricted Burnside Problem (RBP), ∼1940:
  What is order of largest finite quotient $R(d,n)$ of $B(d,n)$, if it exists?

- **Golod-Šafarevič (1964):** answer to GBP is “no”;
  (cf. Ol’shanskii’s Tarski monster)
- **Various authors:** $B(d,n)$ is finite for $n = 2,3,4,6$, but in no other cases with $d > 1$ is it known to be finite; are $B(2,5)$ and $B(2,8)$ finite?
- **Adian (2015):** $|B(d,n)| = \infty$ for odd $n > 100$ and $d \geqslant 2$.
- **Higman-Hall (1956):** reduced (RBP) to prime-power n.
- **Zel’manov (1990-91):** $R(d,n)$ always exists!

Application: Burnside groups

Burnside groups:

- $B(d,n) = \langle x_1, \dots, x_d \mid g^n = 1$ for all words g in $x_1^{\pm}, \dots, x_n^{\pm}\rangle$
- $R(d,n)$ largest finite quotient of $B(d,n)$; exists by Zel'manov

Implementations of the p-quotient algorithm have been used to determine the order and to compute pcps for some of these groups.

Group	Order	Authors
$B(3,4)$	2^{69}	Bayes, Kautsky & Wamsley (1974)
$R(2,5)$	5^{34}	Havas, Wall & Wamsley (1974)
$B(4,4)$	2^{422}	Alford, Havas & Newman (1975)
$R(3,5)$	5^{2282}	Vaughan-Lee (1988); Newman & O'Brien (1996)
$B(5,4)$	2^{2728}	Newman & O'Brien (1996)
$R(2,7)$	7^{20416}	O'Brien & Vaughan-Lee (2002)

Next: What other quotient algorithms exist?

Nilpotent quotients

Task: For an fp group G, compute its nilpotent quotients.

Recall:

- G is nilpotent of class c if $G > \gamma_2(G) > \ldots > \gamma_{c+1}(G) = 1$.
- Each $\gamma_i(G)/\gamma_{i+1}(G)$ is finitely generated and central in $G/\gamma_{i+1}(G)$.
- If G/N is nilpotent of class d, then $\gamma_{d+1}(G) \leqslant N$.

Analogous to the ANUPQ program:

Nickel'94 has developed a nilpotent quotient algorithm: for an fp group G, iteratively compute epimorphisms from G onto $G/\gamma_i(G)$ (given via pcps).

Approach

- Compute $G/\gamma_2(G) = G/G'$; suppose this is d-generated.
- Iteration: assuming an epimorphism $G \to G/\gamma_i(G)$, compute epimorphism $G \to G/\gamma_{i+1}(G)$ by working with a *nilpotent cover* C of $G/\gamma_i(G)$.
- As in ANUPQ, find a consistent pcp of C and enforce relations of G.

Solvable quotients

A group G is solvable if the derived series $G = G^{(0)} \geqslant G^{(1)} \geqslant \ldots$ terminates at 1, where each $G^{(i+1)} = [G^{(i)}, G^{(i)}]$.

Task: For an fp group G, compute its solvable (polycyclic) quotients.

Problems:

- The sections $G^{(i)}/G^{(i+1)}$ might not be finitely generated!
- Consider submodules: choices are involved . . .

Significant work: Wamsley'77, Baumslag-Cannonito-Miller'81, Leedham-Green'84, Plesken'87, Sims'90, Niemeyer'94+'18, Lo'98, *and others . . .*

B-C-M: described algorithm that tests if an fp-group G has polycyclic quotient $G/G^{(k)}$, and if so, determines pcp. **Sims** considered implementation aspects: *"The algorithm is closely related to various attempts to develop constructive versions of the Hilbert basis theorem."* **Lo** extended it (15000 lines of C-code).

Plesken, Wegner, Niemeyer: Compute finite solvable quotients involving a given set of primes, by constructing iterated extensions.

Quotient algorithms

Brief survey

- **abelian:** Smith (1861) / Poincaré (1900) (?)
- **p-quotient:** MacDonald, Newman, Havas, O'Brien, Nickel, Niemeyer, ...
- **nilpotent:** Nickel'94, Sims'94
- **solvable/polycyclic:** Wamsley'77, Baumslag-Cannonito-Miller'81, Leedham-Green'84, Plesken'87, Sims'90, Niemeyer'94, Lo'98
- **simple:** Plesken-Fabiańska'09, Jambor'12-15, Bridson-Evans-Liebeck'19

Plesken, Fabiańska, Jambor developed so-called L_2-algorithms, that attempt finding quotients of an fp group that are isomorphic to $\mathrm{PSL}_2(q)$ or to $\mathrm{PGL}_2(q)$ for some prime power q.

Bridson-Evans-Liebeck considered the question: for which collections of finite simple groups is there an algorithm that determines the members of the collection that are quotients of an arbitrary fp group?

GAP

In GAP: Various functions are available to compute quotients of fp groups, e.g.:

```
gap> GQuotients;; MaximalAbelianQuotient;; EpimorphismPGroup;;
gap> NqEpimorphismNilpotentQuotient;; EpimorphismSolvableQuotient;;
gap> PqEpimorphism;; LargerQuotientBySubgroupAbelianization;;
```

`GQuotients(F,G)` attempts to compute all epimorphisms from an fp group F onto G up to automorphisms of G; maybe use with `SimpleGroupsIterator()`.

`EpimorphismSolvableQuotient(F,n)` attempts to compute an epimorphism from an fp group F onto a solvable group of size at most n.

```
gap> F   := FreeGroup( "a", "b", "c", "d" );;
gap> Fp  := F/ParseRelators(F,"a^2, b^2, c^2, d^2, (a*b)^3,
                (b*c)^4, (c*d)^3, c^a=c, d^a=d");
gap> hom := EpimorphismSolvableQuotient(Fp,300);;
gap> Q   := Image(hom);;
gap> StructureDescription(Q);
"D12"
gap> F   := FreeGroup(["a","b"]);;
gap> G   := F/ParseRelators(F,"a^4,b^4");;
gap> Q   := Pq(G : Prime := 2, ClassBound:=5);
<pc group of size 524288 with 19 generators>
```

Conclusion Lecture 2

Things we have discussed in the second lecture:

- polycyclic presentations, collection, consistency
- abelian quotients
- ANUPQ: p-quotients, p-cover, ...
- briefly: nilpotent and solvable quotients

Questions?

Third lecture

▶ Go to Overview

Recap Lecture 2

Things we have discussed in the second lecture:

- polycyclic groups and presentations
- working in pc groups: collection, consistency
- abelian quotients, abelian invariants
- ANUPQ: p-quotients, p-cover, . . .
- briefly: nilpotent, solvable, and simple quotients

Today: rewriting systems, a bit about automatic groups, and maybe a non-solvable quotient algorithm

Rewriting systems

Rewriting systems

The relations of a consistent pcp

$$\langle g_1,\dots,g_n \mid \forall i:\ g_i^{s_i} = w_i(g_{i+1}\dots,g_n),\ \ \forall i<j:\ g_j^{g_i^{\pm}} = w_{i,j}^{\pm}(g_{i+1},\dots,g_n)\rangle$$

describe rewriting rules

$$g_i^{s_i} \mapsto w_i(g_{i+1}\dots,g_n) \quad \text{and} \quad g_jg_i^{\pm} \mapsto g_i^{\pm}w_{i,j}^{\pm}(g_{i+1},\dots,g_n) \text{ for } i<j.$$

Applying these rules iteratively determines the unique *normal form* of an element.

Example:

The pcp $\langle a,b,c,d \mid a^2=b,\ b^2,\ c^3=d^2,\ d^3,\ c^a=c^2d,\ d^a=d^2\rangle$ yields rules

$$a^2\mapsto b,\quad b^2\mapsto 1,\quad c^3\mapsto d^2,\quad d^3\to 1,\quad ca\mapsto ac^2d,\quad da\mapsto ad^2,$$

and we can rewrite (*collect*)

$$\underline{ca}d^2ca \;\mapsto\; ac^2\underline{dd^2}ca \;\mapsto\; a\underline{c^3}a \;\mapsto\; ad\underline{da} \;\mapsto\; a\underline{da}d^2 \;\mapsto\; a^2\underline{d^3}d \;\mapsto\; \underline{a^2}d \;\mapsto\; bd.$$

Indeed, bd is the normal form of cad^2ca.

Rewriting systems

More generally, a *rewriting system* (Thue 1914) allows working with products from a set of allowable symbols using a set of *simplifying rules*.
A good ref. for the next slides: Book and Otto, "*String Rewriting Systems*", 1993.

A rewriting system $S = (X, \mathcal{R})$ has an alphabet set X and rewriting rules

$$\mathcal{R} \subseteq X^* \times X^*;$$

for $u, v \in X^*$ write $u \to v$ if $u = a\ell b$ and $v = arb$ for some $(\ell, r) \in \mathcal{R}$.

- The reflexive and transitive (and symmetric) closure is denoted $\xrightarrow{*}$ (and $\xleftrightarrow{*}$); note that " $\xleftrightarrow{*}$ " is *not the same* as " $\xrightarrow{*}$ and $\xleftarrow{*}$ ".
- Concatenation of words is a well-defined operation on $\xleftrightarrow{*}$-equivalence classes, and defines the quotient monoid $M = M(X, \mathcal{R})$. It is a group when the class of every $a \in X$ has an inverse $A \in X^*$ (so $aA \xrightarrow{*} 1$, empty word).

For example: Consider $X = \{g, G\}$ and $\mathcal{R} = \{(gG, 1), (Gg, 1)\}$. Then

$$ggGgg\underline{gG} \to g\underline{gG}gg \to ggg = g^3 \quad (\textit{reduced}).$$

In particular, $M(X, \mathcal{R}) \cong C_\infty$ is a group with class reps 1, g^n, and G^n for $n \in \mathbb{N}$.

Rewriting systems: properties

Let $S = (X, \mathcal{R})$ be a rewriting system and $x, y, z \in X^*$.

- If there is no $w \in X^*$ with $x \to w$, then x is reduced.
- If $x \overset{*}{\leftrightarrow} y$ and y is reduced, then y is a normal form for x.
- S is confluent: if $x \xrightarrow{*} y$ and $x \xrightarrow{*} z$, then there is $w \in X^*$ with $y \xrightarrow{*} w$ and $z \xrightarrow{*} w$.
- S is locally confluent: if $x \to y$ and $x \to z$, then there is $w \in X^*$ with $y \xrightarrow{*} w$ and $z \xrightarrow{*} w$.
- S has the Church-Rosser property: if $x \overset{*}{\leftrightarrow} y$, then there is $w \in X^*$ with $x \xrightarrow{*} w$ and $y \xrightarrow{*} w$.

Confluent:

$$\begin{tikzcd} & x \arrow[dl, "*"'] \arrow[dr, "*"] & \\ y \arrow[dr, "*"'] & & z \arrow[dl, "*"] \\ & w & \end{tikzcd}$$

Locally confluent:

$$\begin{tikzcd} & x \arrow[dl] \arrow[dr] & \\ y \arrow[dr, "*"'] & & z \arrow[dl, "*"] \\ & w & \end{tikzcd}$$

Church-Rosser:

$$\begin{tikzcd} x \arrow[rr, leftrightarrow, "*"] \arrow[dr, "*"'] & & y \arrow[dl, "*"] \\ & w & \end{tikzcd}$$

Theorem. S has Church-Rosser property $\iff$ S is confluent.

Proof. "$\Rightarrow$": ✓ For "$\Leftarrow$" use induction on n where $x \overset{n}{\leftrightarrow} y$. For $n = 0$ let $x = y = w$. Now let $x \overset{n+1}{\leftrightarrow} y$. Let x_1 with $x \leftrightarrow x_1 \overset{n}{\leftrightarrow} y$. By the ind. hypothesis, there is w' with $x_1 \xrightarrow{*} w'$ and $y \xrightarrow{*} w'$. If $x \to x_1$, then $x \xrightarrow{*} w'$ and $y \xrightarrow{*} w'$. ✓ If $w' \xleftarrow{*} x_1 \to x$, then confluence show there is w with $x \xrightarrow{*} w \xleftarrow{*} w' \xleftarrow{*} y$. ✓

Rewriting systems: properties

Let $S = (X, \mathcal{R})$ be a rewriting system.

Corollary. If S is confluent, then $x \in X^*$ has at most one normal form.

The relation $\rightarrow$ is noetherian if there is no infinite chain $x_0 \rightarrow x_1 \rightarrow \ldots$ in X^*; it is complete (or *convergent*) if it is noetherian and confluent. **This is desirable!**

Theorem. If S is complete, then every $x \in X^*$ has a unique normal form.

In general, it is undecidable whether S is confluent; however, Thm 1.1.13:

Lemma. Let S be noetherian. Then: S confluent $\iff$ S locally confluent.

If S is notherian, then $x \overset{+}{\rightarrow} y$ defines a special *strict partial order* "$x > y$" (irreflexive, asymmetric, transitive) on X^*. The converse is also true (Thm 2.2.4):

Lemma. S is noetherian $\iff$ there is a strict partial order "$>$" on X^* with

- compatible with the rules: $\ell > r$ for each $(\ell, r) \in \mathcal{R}$,
- well-founded: there is no infinite chain $x_0 > x_1 > \ldots$ in X^*,
- admissible: for all $g, h, x, y \in X^*$, if $g > h$, then $xgy > xhy$.

Undecidable problems

Unfortunately, but expected, many problems are undecidable.

Theorem. The question whether a finite rewriting system is noetherian is undecidable in general.

See Thm 2.5.13 for a reduction to the Halteproblem.

It is also shown that:

Theorem. The question whether a finite rewriting system is (locally) confluent is undecidable in general.

However, if we know that a rewriting system is noetherian, then the previous slide shows that "*confluence* = *local confluence*".

Moreover, with these assumptions, there is an algorithm to verify that!

Checking for local confluence

Let $S = (X, \mathcal{R})$ be noetherian, so "S confluent $\iff$ S locally confluent".

The following is from Section 2.3. If S is *not* locally confluent, then in X^* there exist $c \leftarrow w \rightarrow d$, but c and d have no common reduction.

To check local confluence, it suffices to consider certain *minimal* w coming from overlaps in the rules $\mathcal{R}$: for each pair $(\ell_1, r_1), (\ell_2, r_2) \in \mathcal{R}$ one considers overlaps of ℓ_1 and ℓ_2 and then applies the first or second rule:

- $w = x\ell_1 = \ell_2 y$ with $|x| < |\ell_2|$, yielding the critical pair (xr_1, r_2y);
- $w = \ell_1 = x\ell_2 y$, yielding the critical pair (r_1, xr_2y).

Theorem. S is confluent if and only if the elements of each critical pair can be rewritten (via so-called *left-most reductions*) to the same reduced word.

Thus, one can decide whether a finite noetherian rewriting system is confluent.

Knuth-Bendix

Now also assume "well-ordering": $x > y$ or $x = y$ or $y > x$ for all $x, y \in X^*$.

If a critical pair (c, d) leads to reduced $\hat{c} \neq \hat{d}$, then $S = (X, \mathcal{R})$ is not confluent.

Idea: if $\hat{c} > \hat{d}$, add rule $\hat{c} \to \hat{d}$ to $\mathcal{R}$ to resolve this conflict; otherwise add $\hat{d} \to \hat{c}$. However, this new rule might lead to new critical pairs.

Knuth-Bendix Completion Procedure (KB)

Iterate this process! This produces a sequence $\mathcal{R}_0 = \mathcal{R}$, $\mathcal{R}_1$, $\mathcal{R}_2$, ... of rules. If the critical pairs for $\mathcal{R}_i$ do not produce new rules, then return $S' = (X, \mathcal{R}_i)$.

Comments:

- If KB terminates, it returns a confluent $S' = (X, \mathcal{R}')$ with $\overset{*S}{\leftrightarrow} = \overset{*S'}{\leftrightarrow}$.
- If KB doesn't terminate, then it *enumerates* an infinite confluent rewriting system. Partial results (the systems $\mathcal{R}_i$) can still be useful in practice!
- KB terminates if and only if there is a finite confluent $S' = (X, \mathcal{R}')$ with $\overset{*S}{\leftrightarrow} = \overset{*S'}{\leftrightarrow}$ and $\ell > r$ for each $(\ell, r) \in \mathcal{R}'$.

KB: example

Let $X = \{a, b, c\}$ and $\mathcal{R} = \{(a^2, 1), (b^2, 1), (ab, c)\}$. Let "$<$" be the shortlex ordering induced by $a < b < c$; this is compatible with $\mathcal{R}_0 = \mathcal{R}$.

Critical pairs: consider $b \leftarrow aab \rightarrow ac$ and $cb \leftarrow abb \rightarrow a$, so define

$$\mathcal{R}_1 = \mathcal{R}_0 \cup \{(ac, b), (cb, a)\}.$$

No new critical pairs since all additional overlaps are resolved:

- $c \leftarrow aac \rightarrow ab \rightarrow c$
- $1 \leftarrow bb \leftarrow acb \rightarrow aa \rightarrow 1$
- $c \leftarrow cbb \rightarrow ab \rightarrow c$

Thus, KB produces the confluent rewriting system $(X, \mathcal{R}_1)$.

The ordering can influence whether or not KB terminates (see next slides).

Common orderings are shortlex (wrt some total ordering on X) or the wreath product ordering (see problem sheet).

KB: GAP

Holt's software KBMAG (Knuth-Bendix on Monoids and Automatic Groups) can be used in GAP.

Here is our first example in GAP:

```
gap> LoadPackage("kbmag");
true
gap> FM := FreeMonoid(3);;
gap> a  := FM.1;; b:=FM.2;; c:=FM.3;; id := Identity(FM);;
gap> M  := FM/[ [a^2,id], [b^2,id], [a*b,c] ];;
gap> R  := KBMAGRewritingSystem(M);
rec(isRWS:=true, silent:=true, generatorOrder:=[_m1,_m2,_m3], inverses:=[,,],
ordering:="shortlex", equations:=[[_m1^2,IdWord],[_m2^2,IdWord],[_m1*_m2,_m3]])
# m1, m2, m3 correspond to a, b, c
gap> MakeConfluent(R);;
gap> R;
rec(isRWS:=true, isConfluent:=true, silent:=true, inverses := [,,],
generatorOrder:=[_m1,_m2,_m3], ordering:="shortlex",
equations := [[_m1^2,IdWord],[_m2^2,IdWord],[_m1*_m2,_m3],
[_m1*_m3,_m2],[_m3*_m2,_m1]]) # these are our new rules ac->b, cb->a
gap> ReducedWord(R,a*b*a*b^2);
m3*m1 # this is c*a
```

KB: GAP

The second example is the free abelian group of rank 2, given as $(X,\mathcal{R})$, where $X=\{a,b,\bar{a},\bar{b}\}$. With $a<\bar{a}<b<\bar{b}$, the rules are (ba,ab) and $(a\bar{a},1)$, $(\bar{a}a,1)$, $(b\bar{b},1)$, $(\bar{b}b,1)$, $(b\bar{a},\bar{a}b)$, $(\bar{b}a,a\bar{b})$, $(\bar{b}\bar{a},\bar{a}\bar{b})$, and we quickly get:

```
gap> F := FreeGroup(2);; a:=F.1;; b:=F.2;; G := F/[b*a/(a*b)];;
gap> R := KBMAGRewritingSystem(G);;
gap> MakeConfluent(R);;
gap> R;
rec((...), generatorOrder:=[_g1,_g2,_g3,_g4], isConfluent:=true,
equations:=[[_g1*_g2,IdWord],[_g2*_g1,IdWord],[_g3*_g4,IdWord],[_g4*_g3,IdWord],
[_g3*_g1,_g1*_g3],[_g4*_g1,_g1*_g4],[_g3*_g2,_g2*_g3],[_g4*_g2,_g2*_g4]])
```

For $\bar{b}<a<\bar{a}<b$ and adjusted $\mathcal{R}$, we get $a\leftarrow\bar{b}ba\rightarrow\bar{b}ab$ and $\bar{a}\leftarrow\bar{b}b\bar{a}\rightarrow\bar{b}\bar{a}b$.

```
gap> F := FreeGroup(2);; a:=F.1;; b:=F.2;; G := F/[b*a/(a*b)];;
gap> R := KBMAGRewritingSystem(G);;
gap> ReorderAlphabetOfKBMAGRewritingSystem(R, (1,2,3,4));;
gap> R;
rec((...), generatorOrder:=[_g4,_g1,_g2,_g3],equations := [[_g3*_g1,_g1*_g3]])
gap> MakeConfluent(R);; # does not terminate
```

KB enumerates $\mathcal{R}_0,\mathcal{R}_1,\ldots$ where $\mathcal{R}_i=\mathcal{R}\cup\{(\bar{b}a^jb,a^j),(\bar{b}\bar{a}^jb,\bar{a}^j):j=1,\ldots,i\}$.

KB application: verifying polycyclic

It is undecidable whether an fp group $G = \langle X \mid \mathcal{R} \rangle$ is pc. However, if G is pc, then KB can assist to verify it. I'll skip the details here (see Sims §11.8 for more information); below is a sketch.

If G is polycyclic with derived length k, then with the pc quotient algorithm, one can determine a consistent pcps for $Q_i = G/G^{(i)}$ for $i = 1, 2, \ldots$ until $Q_k = Q_{k+1}$; thus, k can be determined.

Let $X = \{x_1, \ldots, x_r\}$. Use the epimorphism $G \to Q_k$ to introduce additional generators $x_{r+1}, \ldots, x_t$ that map to a pcgs of Q_k. Let $\mathcal{R}'$ be the relations in $\mathcal{R}$ together with relations defining $x_{r+1}, \ldots, x_t$ in terms of X.

Note: If G is pc, then $Q_k = G$ and $(x_1, \ldots, x_t)$ is a pcgs of G.

Now start KB on G with input $x_1, \ldots, x_t$ and $\mathcal{R}'$ and wreath-product ordering defined by $x_1^{-1} > x_1 > x_2^{-1} > x_2 > \ldots > x_t^{-1} > x_t$.

If *run long enough*, KB will terminate and output a pcp for G.

KB application: verifying isomorphism

Let $G = \langle x_1, \ldots, x_n \mid \mathcal{R} \rangle$ and $H = \langle y_1, \ldots, y_m \mid \mathcal{S} \rangle$ be fp groups. It is undecidable whether $G \cong H$. However, if $G \cong H$, then KB can assist to prove it. I'll skip the details here (see Holt-Rees (1992)); below is a sketch.

Let θ be the map that assigns each x_i to a word w_i in $\{y_1, \ldots, y_m\}$.

By von Dyck, θ extends to a homomorphism $G \to H$ if and only if each relator $r = x_{i_1}^{e_1} \ldots x_{i_\ell}^{e_\ell}$ in $\mathcal{R}$ satisfies

$$\theta(r) = w_{i_1}^{e_1} \ldots w_{i_\ell}^{e_\ell} = 1_H.$$

Idea: Running KB on H might be able to establish that each $\theta(r) = 1$ in H; for this it might not be necessary that KB terminates.

If θ is a homomorphism, use KB to generate all reduced words in G up to a certain length, map them to H, and check if their reductions cover all the generators of H; then θ is an epimorphism.

Dehn's rewriting system

Let $G = \langle X \mid \mathcal{R}\rangle$ be an fp group with $\mathcal{R}$ cyclically reduced.
Denote the set of all $r \in \mathcal{R}$ and their cyclic shifts and inverses by $\hat{\mathcal{R}}$.

If $r \in \hat{\mathcal{R}}$ is factorised as $r = ab$, then $a = b^{-1}$ in G, which yields a rule $a \mapsto b^{-1}$.
This is the idea for the Dehn rewriting system:

Dehn RWS Write each $r \in \hat{\mathcal{R}}$ as $r = ab$ with $|a| > |b|$ and define rules $a \to b^{-1}$. The Dehn rewriting system for $\langle X \mid \mathcal{R}\rangle$ is $S = (X, \mathcal{D})$ where $\mathcal{D}$ is the set of these rules together with $xx^{-1} \to 1$ for $x \in X \cup X^{-1}$.

Definition: The group G has a Dehn presentation if the Dehn RWS solves the word problem (Dehn's algorithm), that is, *if every $w \in F_X$ (reduced in F_X) with $w =_G 1$ has a subword that is more than half of some defining relator*.

Greendlinger's lemma (Lyndon-Schupp (1977), p. 247)
If $G = \langle X \mid \mathcal{R}\rangle$ is an fp group with $\mathcal{R}$ cyclically reduced, then Dehn's algorithm solves the word problem if G is $C'(1/6)$, or $C'(1/4)$ and $T(4)$, or ...

Small cancellation: A subword w is a piece if there are $u \neq v$ with $wu, wv \in \hat{\mathcal{R}}$, and $\langle X \mid \mathcal{R}\rangle$ is $C'(\lambda)$ if whenever w is a piece in $r \in \hat{\mathcal{R}}$, then $|w| < \lambda |r|$.

Dehn's algorithm: example

Let $G = \langle X \mid \mathcal{R} \rangle$ where $X = \{a, b, c, d\}$ and

$$\mathcal{R} = \{a^7,\ b^7,\ d^7,\ ab^{-1}\underline{abc^{-1}d^{-1}c}d\}.$$

Pieces are $a^{\pm 1}$, $b^{\pm 1}$, $c^{\pm 1}$, $d^{\pm 1}$, so G is $C'(1/6)$.

Consider $w = a^4b^6abc^{-1}d^{-1}cd^{-6}a^4 w = a^4b^6\underline{abc^{-1}d^{-1}c}d^{-6}a^4$ and apply Dehn's algorithm.

We have $\underline{abc^{-1}d^{-1}c}dab^{-1} \in \hat{\mathcal{R}}$, so there is a rule $abc^{-1}d^{-1}c \to ba^{-1}d^{-1}$:

$$w = a^4b^6abc^{-1}d^{-1}cd^{-6}a^4 \quad \to \quad a^4b^6ba^{-1}d^{-1}d^{-6}a^4 = a^4b^7a^{-1}d^{-7}a^4.$$

Since $a^7, b^7, d^7 =_G 1$, we see that $w =_G 1$; however, Dehn's algorithm continues as follows: $b^7, d^{-7}, a^7 \in \hat{\mathcal{R}}$ yield rules $b^4 \to b^{-3}$, $d^{-4} \to d^3$, $a^4 \to a^{-3}$, and so

$$w \quad \xrightarrow{*} \quad a^4b^7a^{-1}d^{-7}a^4 = a^4b^4b^3a^{-1}d^{-4}d^{-3}a^4 \quad \to \quad a^7 = a^4a^3 \quad \to \quad 1.$$

Dehn's algorithm

Groups with $C'(1/6)$ are hyperbolic, as defined by Gromov 1987: geodesic triangles in the Cayley graph are *thin*. "Most fp groups are hyperbolic".

Hyperbolic groups are exactly those that have a finite Dehn presentation. Deciding whether an fp group is hyperbolic is undecidable in general, but if the group is hyperbolic, then one can in principle verify it.

(See Epstein-Holt (2001) and Alonso-Brady-Cooper-*et al.* (1991) for details and references.)

Computational tools:

- Pfeiffer's GAP package 'walrus' provides an implementation that attempts to prove that a given fp group is hyperbolic, and if so, to calculate a rewriting system for a Dehn algorithm.
- Hyperbolic groups are automatic groups (see later), and KBMAG can compute with these and attempt to verify hyperbolicity. According to Holt, it's slower than the above method, but it succeeds on more examples.

Dehn's algorithm: GAP

Here is our example in KBMAG.

Currently, KBMAG cannot apply Dehn's algorithm directly, but it can reduce words to normal forms in automatic groups, and $C'(1/6)$ groups are automatic:

```
gap> LoadPackage("kbmag");;
gap> F:=FreeGroup(["a","b","c","d"]);
<free group on the generators [ a, b, c, d ]>
gap> G:=F/ParseRelators(F,"a^7,b^7,d^7, a*b^-1*a*b*c^-1*d^-1*c*d ");;
gap> R:=KBMAGRewritingSystem(G);;
gap> AutomaticStructure(R); # find automatic structure, see next slide
true
gap> AssignGeneratorVariables(F);
#I  Assigned the global variables [ a, b, c, d ]
gap> w:=a^4*b^6*a*b*c^-1*d^-1*c*d^-6*a^4;;
gap> ReducedWord(R,w);
<identity ...>
```

Dehn's algorithm: Magma

We haven't really talked about Magma (magma.maths.usyd.edu.au/magma), but this system also provides functions to deal with hyperbolic/automatic groups.

```
> F<a,b,c,d> := FreeGroup(4);
> rels := [a^7,b^7,d^7, a*b^-1*a*b*c^-1*d^-1*c*d];
> ishyp, isdehn, Dehn := IsHyperbolic(F,[],rels);
> ishyp, isdehn;
true true /* group is hyperbolic, Dehn algorithm has been computed */
> w := a^4*b^6*a*b*c^-1*d^-1*c*d^-6*a^4;
> IsIdentity(w,Dehn);
true Id(F)
```

Based on Holt, Linton, Neunhöffer, Parker, Pfeiffer, Roney-Dougal (2021).

Next: A quick look at automatic groups.

finite state automaton

A finite state automaton (fsa) is $M = (S, A, \tau, F, s_0)$ where

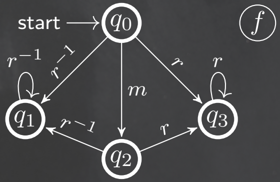
- A alphabet, S set of states with $F \subseteq S$ accept states and $s_0 \in S$ initial state,
- $\tau\colon S \times A \to S$ is a transition function.

A word $a_0 \ldots a_n \in A^*$ is accepted by M if $\tau(\cdots\tau(\tau(s_0, a_0), a_1), \ldots, a_n) \in F$. Accepted language: $L(M) = \{w \in A^* : w \text{ accepted by } M\}$ (*regular languages*).

Example. $M = (S, A, \tau, F, s_0)$ where

- $S = \{f, q_0, \ldots, q_3\}$ and $A = \{r, r^{-1}, m, m^{-1}\}$,
- τ as in diagram, and "$\to f$ " (fail) if not listed,
- $F = \{q_0, \ldots, q_3\}$ and $s_0 = q_0$.

$L(M) = \{m^a r^b : a \in \{0, 1\}, b \in \mathbb{Z}\}$ maps onto the normal words in $D_\infty = \mathrm{Pc}\langle r, m \mid r^2, r^m = r^{-1}\rangle$.



Another example: A complete coset table with $s_0 = 1$ and $F = \{i\}$; the accepted language is the set of words representing elements in coset i.

fsa: applications in KB

In KB, a fsa can detect if a word is reduced in a rewriting system $S = (X, \mathcal{R})$.
(See Handbook §13.1.3 or Sims §3.5 for details and references.)

Suppose that if $(\ell, r) \in \mathcal{R}$, then r and proper substrings of ℓ are reduced.
Now let M_S be the following (partial) fsa with

- states P are all prefixes of ℓ for all $(\ell, r) \in \mathcal{R}$;
- alphabet is X; initial state is ε;
- accept states are all *proper* prefixes of ℓ for all $(\ell, r) \in \mathcal{R}$;
- non-accept states are ℓ for all $(\ell, r) \in \mathcal{R}$; they are dead;
- if $s \in P$ is not dead, the transition $\tau(s, x)$ is the longest suffix of sx in P.

If $w = w_1 \ell w_2 \in X^*$ and ℓ is the first occurrence of a left-hand side in $\mathcal{R}$, then after reading $w_1 \ell$ the fsa M_S is in dead state $\ell \in P$.

Thus: The accepted language $L(M_S)$ is the set of reduced words.

Also: If S is confluent and defines a group G, then M_S is a word-acceptor for G: for each $g \in G$, the fsa M_S accepts at least one (reduced) word representing g.

automatic groups

Informally, an automatic group is a finitely generated group for which finite state automata accept a language representing group elements (normal forms) and decide whether two such normal forms represent group elements that differ by right multiplication by a single generator. This is called an automatic structure.

The theory of automatic groups evolved in the mid-1980's due to work of Cannon, Thurston, Epstein, Holt, Paterson, Levy (*Word processing in groups*, 1992).
With an automatic structure one can:

- produce unique "normal forms" (quadratic algorithm),
- enumerate group elements (the first n can be listed in time $O(n \log n)$).

Automatic groups include: hyperbolic, virtually abelian, Coxeter, and Braid groups; direct and free products of automatic groups, subgroups and supergroups of finite index in automatic groups, . . .

However: automatic nilpotent groups are virtually abelian.

automatic groups: definition

Skipping some details, the formal definition is ...

A group G with (monoid) generators X is automatic if there are

- a fsa W (*word acceptor*), that is, $L(W)$ maps onto G, and
- a 2-*variable fsa* M_x (*multiplier automaton*) for each $x \in X \cup \{\varepsilon\}$ such that for all $u, v \in L(W)$ we have

  $$(u, v)^+ \in L(M_x) \iff u =_G vx,$$

  that is, M_x recognises multiplication on the right by x in $L(W)$.

Algorithms often require the following: The group is shortlex automatic if W accepts precisely the minimal words under some shortlex ordering "$\leqslant$" of X^*.

You'll see examples on the problem sheet; here we focus on example computations.

Note: Being automatic is a property of G, not of X; it is unknown whether every automatic group is shortlex automatic for some ordering.

How to get an automatic structure? Not here ... (see Epstein et al., 35 pages).

automatic groups: examples

Let's look at the *Heineken group*

$$G = \langle a, b, c \mid [a,[a,b]] = c,\ \ [b,[b,c]] = a,\ \ [c,[c,a]] = b\rangle.$$

It was proposed by Heineken as a possible example of a finite group with a balanced symmetrical presentation. This was motivated by the fact that $\langle x, y, z \mid [x, y] = z, [y, z] = x, [z, x] = y\rangle \cong 1$, and G has a quotient of size $60 \cdot 2^{24}$.

```
gap> LoadPackage("kbmag");
gap> F := FreeGroup( "a", "b", "c" );;
gap> G := F/ParseRelators(F,"[a,[a,b]]*c^-1, [b,[b,c]]*a^-1, [c,[c,a]]*b^-1");;
gap> R := KBMAGRewritingSystem(G);
gap> SetInfoLevel( InfoRWS, 1 );
gap> MakeConfluent(R);
#I  Calling external Knuth-Bendix program.
#Maximum number of equations exceeded.
#Halting with 32767 equations.
#I  External Knuth-Bendix program complete.
#I  System computed is NOT confluent.
false
```

automatic groups: examples

Let's look at the group

$$G = \langle a, b, c \mid [a,[a,b]] = c,\ \ [b,[b,c]] = a,\ \ [c,[c,a]] = b \rangle.$$

Now let's try to compute an automatic structure.

```
gap> ResetRWS(R); # reset rws because we ran KB already
gap> AutomaticStructure(R);
#I  Calling external automatic groups program.
#Running Knuth-Bendix Program: (...)
#Maximum number of states exceeded.
#Halting with 165 equations.
(...)
#Knuth-Bendix program failed or was inconclusive. Giving up.
#I  Computation was not successful.
false
```

automatic groups: examples

We look at $G = \langle a, b, c \mid [a,[a,b]] = c,\ [b,[b,c]] = a,\ [c,[c,a]] = b \rangle$.

```
gap> AutomaticStructure(R,true); # set "large" to true
#I  Calling external automatic groups program.
(...)
#Word-acceptor with 1167 states computed.
#General multiplier with 2973 states computed.
#Validity test on general multiplier succeeded.
#Running program to verify axioms on the automatic structure
#General length-2 multiplier with 3251 states computed.
#Checking inverse and short relations.
(...)
#Axiom checking succeeded.
#I  Computation was successful - automatic structure computed.
#Minimal reducible word acceptor with 1428 states computed.
#Minimal Knuth-Bendix equation fsa with 2743 states computed.
#Correct diff1 fsa with 407 states computed.
#Correct diff2 fsa with 407 states computed.
true
gap> Size(R);
infinity # infinitely many normal forms, so G is infinite
gap> List(GeneratorsOfGroup(F),x -> Order(R,x));
[ infinity, infinity, infinity ]
```

automatic groups: examples

We still look at $G = \langle a, b, c \mid [a,[a,b]] = c,\ [b,[b,c]] = a,\ [c,[c,a]] = b \rangle$.

```
gap> AssignGeneratorVariables(F); # need to work with gens of free group
gap> w  := b*a^2*b^-1*a*b^5*c^-1*a^-2;
gap> wr := ReducedWord(R,w);
a*b*(a*c)^2*b^4*c^-1*a^-2
gap> ReducedWord(R,w*wr^-1);
<identity ...>
gap> EnumerateReducedWords(R, 1, 2);
[ a, a^2, a*b, a*b^-1, a*c, a*c^-1, a^-1, a^-2, a^-1*b, a^-1*b^-1, a^-1*c,
  a^-1*c^-1, b, b*a, b*a^-1, b^2, b*c, b*c^-1, b^-1, b^-1*a, b^-1*a^-1, b^-2,
  b^-1*c, b^-1*c^-1, c, c*a, c*a^-1, c*b, c*b^-1, c^2, c^-1, c^-1*a, c^-1*a^-1,
  c^-1*b, c^-1*b^-1, c^-2 ]
gap> Size( EnumerateReducedWords(R, 1, 7));
112914
```

automatic groups: cosets

Let G be an fp group with subgroup $H \leqslant G$ given by generators. KBMAG can also perform its operations with the cosets of H (rather than the elements of G): the words in reduced form then correspond to *minimal representatives* under the ordering of the system of the right cosets of H in G.

If run successfully: the index $[G : H]$, a confluent rewriting system, and a presentation for H can be computed.

The following example (also from the KBMAG manual) considers the Fibonacci group $G = F(8,2)$ and a subgroup $U \leqslant G$ defined as follows:

$$G = \langle a,b,c,d,e,f,g,h \mid abc^{-1}, bcd^{-1}, cde^{-1}, def^{-1}, efg^{-1}, fgh^{-1}, gh^{-1}a, hab^{-1}\rangle$$

with subgroup

$$U = \langle a, e\rangle \leqslant G.$$

automatic groups: cosets

From the last slide: Consider the subgroup $U = \langle a, e \rangle \leqslant G$ of

$$G = \langle a, b, c, d, e, f, g, h \mid abc^{-1}, bcd^{-1}, cde^{-1}, def^{-1}, efg^{-1}, fgh^{-1}, gh^{-1}a, hab^{-1} \rangle.$$

```
gap> F := FreeGroup(["a","b","c","d","e","f","g","h"]);;
gap> G := F/ParseRelators(F,"a*b=c,b*c=d,c*d=e,d*e=f,e*f=g,f*g=h,g*h=a,h*a=b");;
gap> R := KBMAGRewritingSystem(G);;
gap> AssignGeneratorVariables(F);;
gap> U := SubgroupOfKBMAGRewritingSystem(R, [a,e]);;
gap> AutomaticStructureOnCosetsWithSubgroupPresentation(R,U,true);; # true
gap> P := PresentationOfSubgroupOfKBMAGRewritingSystem(R,U); # U as fp grp
<fp group of size infinity on the generators [ f1, f3 ]>
gap> RelatorsOfFpGroup(P);
[  ]
gap> Index(R,U);
infinity # so U is a free group of rank 2, with [G:U] = infinity
gap> w := a^-2*f^-1*h^2*d^2*e^-1*a^-2;
gap> IsReducedCosetRepresentative(R,U,w);
false
gap> wred := ReducedCosetRepresentative(R,U,w);
g^-1
gap> List([a,b,c,d,e,f,g,h],x -> ReducedCosetRepresentative(R,U,x));
[ <identity ...>, b, b, d, <identity ...>, b^-1, b^-1, h ]
```

Non-solvable quotients

▶ end

Non-solvable quotients

This part is joint work with Alexander Hulpke (2022).

Given set-up:

- Epimorphism $\varphi\colon G \to H$ from an fp group G onto finite group H.
- G and H are e-generated; images of a free group F of rank e.
- H is *not* assumed to be solvable.

Aim:

- Lift φ to a larger quotient $\tilde{\varphi}\colon G \to \tilde{H}$.
- We will assume that $\ker \pi$ is $\mathbb{Z}_p H$-module.[4]

$$\begin{tikzcd}
F \arrow[r] \arrow[d] & \hat{H}_{p,e} \arrow[d] \\
G \arrow[r, "\tilde{\varphi}"] \arrow[dr, "\varphi"'] & \tilde{H} \arrow[d, "\pi"] \\
 & H
\end{tikzcd}$$

Idea: Use suitable cover $\hat{H}_{p,e}$, then evaluate relators of G.

▸ results
▸ example
▸ end

[4] In the solvable radical paradigm, non-solvable bits show up "on top": $A \geqslant \mathrm{PKer}(A) \geqslant \mathrm{Soc}^*(A) \geqslant O_\infty(A) \geqslant 1$; see Seress's book, Section 6.1 + references therein.

The cover $\hat{H}_{p,e}$

Theorem (based on Gaschütz: p-cover of rank e)

For finite, e-generated $H = F/M$ let $\hat{H}_{p,e} = F/M_p$ where $M_p = [M,M]M^{[p]}$.

- $\hat{H}_{p,e}$ is a finite e-gen. extension of H by $\mathbb{Z}_p H$-module $\mathcal{M}_{H,p,e} = M/M_p$.
- $\hat{H}_{p,e}$ maps onto every finite e-gen. extension E of H by a $\mathbb{Z}_p H$-module.
- The isomorphism types of $\hat{H}_{p,e}$ and $\mathcal{M}_{H,p,e}$ depend only on H, e, p.

Our proof depends on results of **Gaschütz'54**, who also investigated the *p-representation module* M/M_p ...

$$\begin{tikzcd}
\hat{H}_{p,e} \arrow[dd] \arrow[dr, dotted, "\exists"] & \\
& \forall E \arrow[dl] \\
H &
\end{tikzcd}$$

▸ results

OK in theory, but in practice...?

Construction via homomorphisms

Problem: For e-gen. $H = F/M$, get $\hat{H}_{p,e} = F/M_p$ and $\mathcal{M}_{H,p,e} = M/M_p$.

We found an explicit construction as images under a homomorphism.

“Fox construction”

Let $|H| = m$ and let $\psi\colon F \to H$ with $\ker\psi = M$, where F is free of rank e. Then

$$\hat{H}_{p,e} \cong \Psi(F) \quad \text{and} \quad \mathcal{M}_{H,e,p} \cong \Psi(M)$$

where $\Psi = \psi_1 \times \ldots \times \psi_e\colon F \to (H \ltimes \mathbb{Z}_p^m)^e$ is a *specific homomorphism*.

Don’t read now: let F be free on $\{x_1, \ldots, x_e\}$; let $\mathbb{Z}_p^m$ be the regular $\mathbb{Z}_p H$-module. Then each

$$\psi_i\colon F \to H \ltimes \mathbb{Z}_p^m, \quad \psi_i(x_j) = \begin{cases} (\psi(x_j), (0,0,\ldots,0)) & \text{if } i \neq j \\ (\psi(x_i), (1,0,\ldots,0)) & \text{if } i = j. \end{cases}$$

Let $\partial_i\colon F \to \mathbb{Z}F$ be defined by $x_i \mapsto 1$ and $1_F \to 0$ and $\partial_i(uv) = (\partial_i u)v + \partial_i(v)$; these maps are called **Fox derivatives**, see Fox (1953). Our proof uses that

$$\partial = (\psi \circ \partial_1) \times \ldots \times (\psi \circ \partial_e)\colon F \to (\mathbb{Z}H)^e.$$

maps $v \in F$ to 0 iff $v \in [M, M]$, see Johnson (1997).

Make it practical

Problem: $\hat{H}_{p,e}/\mathcal{M} \cong H$ where $\mathcal{M} = \mathcal{M}_{H,p,e}$, but $\mathcal{M}$ has size $p^{1+(e-1)|H|}$.

V-homogeneous cover

For a simple $\mathbb{Z}_p H$-module V, let $\mathcal{M}/V(\mathcal{M})$ be the largest V-homogeneous quotient (direct sum of copies of V). The group

$$\hat{H}_{V,e} = \hat{H}_{p,e}/V(\mathcal{M})$$

maps onto every e-generated extension of H by a V-homogeneous $\mathbb{Z}_p H$-module.

Nice:

- If H is finite p-group, then $\hat{H}_{\mathbf{1},\mathsf{rank}(H)} = H^*$ is the p-cover.
- Can get $\hat{H}_{V,e}$ via a homomorphism Ψ_V, *avoiding* the large module $\mathcal{M}$.

Don't read now: In our "Fox construction", replace each $\psi_i \colon F \to H \ltimes \mathbb{Z}_p^m$ by

$$\psi_i' \colon F \to H \ltimes V^r \quad \text{where} \quad V^r \cong \mathbb{Z}_p^m / V(\mathbb{Z}_p^m)$$

is a cyclic $\mathbb{Z}_p H$-module; now use $\Psi_V = \psi_1' \times \ldots \times \psi_e',\ F \to (H \ltimes V^r)^e$ instead of Ψ. One can determine r and a cyclic generator $z \in V^r$ *directly*, and use these to define ψ_i'.

Make it practical

We don't have $\hat{H}_{V,e} = \Psi_V(F)$, but almost ... We also need the following:

Second Cohomology

Let $\gamma_1, \ldots, \gamma_k$ be 2-cocycles that induce a basis of $H^2(H,V)$; define extensions

$$E(\gamma_i) = \{(h,v) : h \in H, v \in V\} \quad \text{with} \quad (h,v)(g,w) = (hg, v^g w \gamma_i(h,g)),$$

and (surjective) homomorphisms $\varrho_i \colon F \to E(\gamma_i)$ via $x_j \mapsto (\psi(x_j), 1)$.

Now define the homomorphism $\varrho = \Psi_V \times \varrho_1 \times \ldots \varrho_k$, so

$$\boldsymbol{\varrho} \colon F \to \Psi_V(F) \times E(\gamma_1) \times \ldots \times E(\gamma_k)$$

Main result

The largest quotient of $\hat{H}_{p,e}$ that maps to H with V-homogeneous kernel can be constructed as

$$\varrho(F) \cong \hat{H}_{V,e}.$$

Results

We need to compute with H, but H is not (necessarily) solvable.

A reasonable assumption is that we have confluent rewriting system for H.

New(ish): *Hybrid representation*[5] of finite polycyclic-by-finite extensions $V.H$: store rewriting system for H, pcp for V, *tails* in V, and normal form routine.

New(ish): Algorithm to compute $H^2(H,V)$ for such hybrid groups H, using ideas of Holt (1985), Schmidt (2008-10); analogous to the approach for pc groups.

New: A new quotient algorithm that lifts $G \to H$ (with G and H as before) to larger quotients of G.

[5] For related work, see also:

- Baumslag-Cannonito-Robinson-Segal (1991): polycyclic-by-finite (theoretical algorithms)
- Sinanan-Holt (2017): polycyclic-by-finite represented via perm rep and pcp (practical); *"Polycyclic-by-finite groups form the largest known section-closed class of fp groups."*

GAP example

Let's reconsider the Heineken Group

$$G = \langle x, y, z \mid [x,[x,y]] = z,\ [y,[y,z]] = x,\ [z,[z,x]] = y \rangle$$

with $\varphi\colon G \to H = \mathrm{Alt}_5$ defined by $\varphi(x) = (1,2,4,5,3)$ and $\varphi(y) = (1,2,3,4,5)$.

The function LiftQuotientHybrid$(\varphi, 2)$ returns $\tau\colon G \to K$ such that $\ker\varphi/\ker\tau$ is the largest 2-semisimple module quotient of $\ker\varphi$.

```
gap> F   := FreeGroup("x","y","z");;
gap> G   := F/ParseRelators(F,"[x,[x,y]]=z,[y,[y,z]]=x,[z,[z,x]]=y");;
gap> G   := SimplifiedFpGroup(G);;
gap> H   := AlternatingGroup(5);;
gap> phi := GroupHomomorphismByImages(G,H,[G.1,G.2],[(1,2,4,5,3),(1,2,3,4,5)]);;
gap> tau := LiftQuotientHybrid(phi,2);;
gap> K   := Image(tau);;
gap> # now compute kernel of projection K -> H
gap> psi := GroupHomomorphismByImages(K,H,[K.1,K.2],[(1,2,4,5,3),(1,2,3,4,5)]);;
gap> StructureDescription(Kernel(psi));
"C2 x C2 x C2 x C2 x C2" # so K is an extension of H=Alt5 by 2^5
```

GAP example

Recall

$$G = \langle x, y, z \mid [x,[x,y]] = z,\ [y,[y,z]] = x,\ [z,[z,x]] = y \rangle\,,$$

and we started with an epimorphism $\varphi\colon G \to \mathrm{Alt}_5$.

An iteration yields the following quotients of G that extend φ:

extension	seconds
$2^5.\mathrm{Alt}_5$	< 1
$2.2^5.\mathrm{Alt}_5$	2
$2^4.2.2^5.\mathrm{Alt}_5$	2
$2^4.2^4.2.2^5.\mathrm{Alt}_5$	6
$2^2.2^4.2^4.2.2^5.\mathrm{Alt}_5$	18
$2^4.2^2.2^4.2^4.2.2^5.\mathrm{Alt}_5$	51
$2^4.2^4.2^2.2^4.2^4.2.2^5.\mathrm{Alt}_5$	118

This is the quotient of G of size $60 \cdot 2^{24}$ mentioned before.

The end!

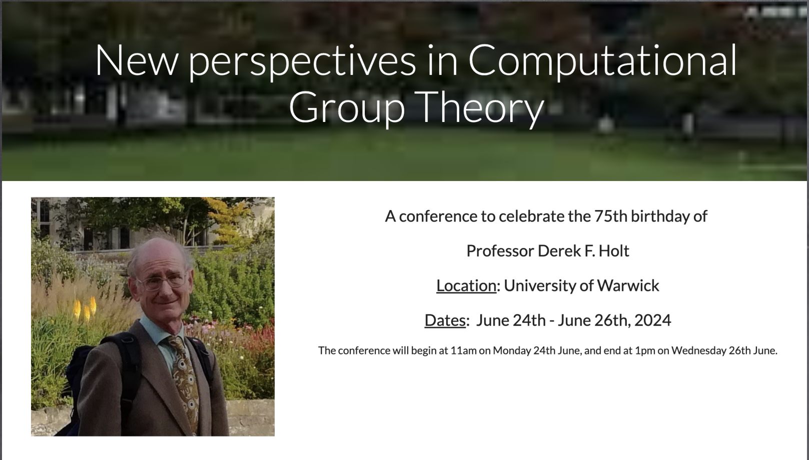
This completes our classes on computing with fp groups!

Looking back . . .

- group presentations (fp, polycyclic, Dehn problems)
- Tietze transformations
- von Dyck, coset enumeration
- low index subgroups, Reidemeister-Schreier
- quotient algorithms (abelian, p-group, (solvable, non-solvable))
- rewriting systems, Knuth-Bendix, Dehn algorithm
- fsa in KB; automatic groups

Thank you for your attention!

. . . and special thanks to Derek Holt for all his contributions!



New perspectives in Computational Group Theory

A conference to celebrate the 75th birthday of

Professor Derek F. Holt

Location: University of Warwick

Dates: June 24th - June 26th, 2024

The conference will begin at 11am on Monday 24th June, and end at 1pm on Wednesday 26th June.