Calculations In Infinite Matrix Groups Using Congruence Images

Alexander Hulpke
Department of Mathematics
Colorado State University
Fort Collins, CO, 80523, USA
www.math.colostate.edu/~hulpke

Joint work with **Alla Detinko** and **Dane Flannery**,

Partially supported by
NSF-DMS #1720146

June 2021

Setup

$$F \xrightarrow{\psi} \Gamma \xrightarrow{\varphi_m} A_m$$

Let Γ be discrete subgroup of Lie Group,

In particular, SL or SP over $\mathbb{Z}$ and its friends (rings of integers, localizations).

$G \leq \Gamma$ given by finite set of generating matrices.
Can we compute the index $[\Gamma : G]$?

Toolset: Computational Group Theory,

Isomorphism $\psi \colon F \to \Gamma$ with F finitely presented explicitly given. Reduction φ_m modulo integer m.

Algorithmic Tools

Finitely presented groups:

Coset Enumeration

Reidemeister-Schreier

Quotient Algorithms

Knuth-Bendix

Matrix groups over finite fields:

Composition tree

Homomorphisms

Finite group calculations

General Paradigms:

Groups given by small set of generators

Orbit algorithm, Schreier Generators.

Subgroup chain data structures for decomposition

Infinity representable by finite information only

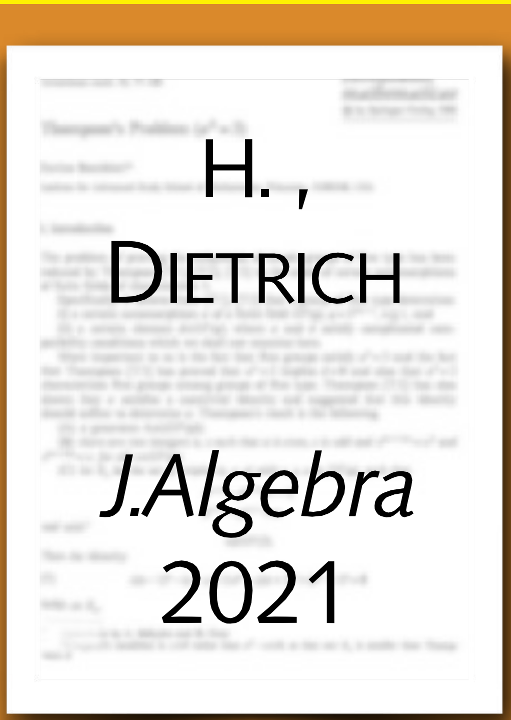
Algorithmic Tools

Finitely presented groups:
Coset Enumeration
Reidemeister-Schreier
Quotient Algorithms
Knuth-Bendix

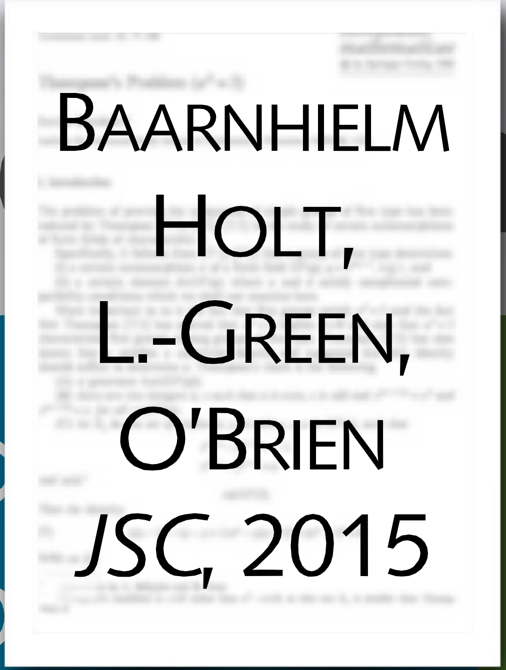
Matrix groups over finite fields:
Composition tree
Homomorphisms
Finite group calculations

General Paradigm
Groups given by s
Orbit algorithm, S
Subgroup chain d
Infinity representa

Shameless Advertisement:
Nonsolvable (` Hybrid")
quotient algorithm.
(Implemented!)


H. ,
DIETRICH
J.Algebra
2021

Alg ic Tools



BAARNHIELM

HOLT,

L.-GREEN,

O'BRIEN

JSC, 2015

Finitely presente

Coset Enumeratio

Reidemeister-Schreier

Quotient Algorithms

Knuth-Bendix

Matrix groups over finite ~~fields~~ Rings:

Composition tree

Homomorphisms

Finite group calculations

General Paradigms:

Groups given by small set of generators

Orbit algorithm, Schreier Generators.

Subgroup chain data structures for decomposition

Infinity representable by finite information only

Algorithmic Tools

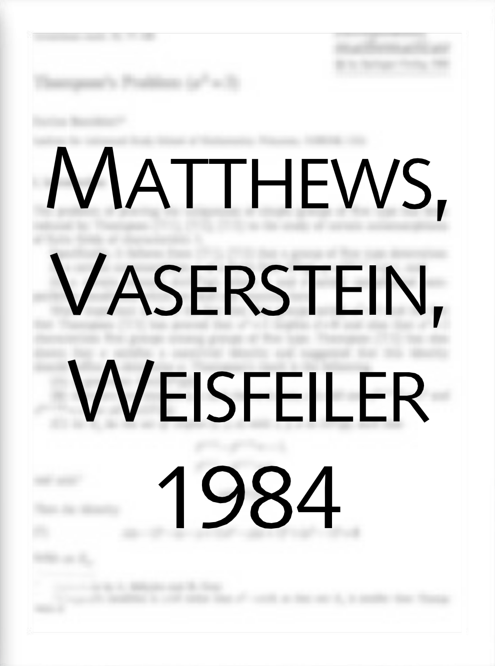
Finitely presented groups:
Coset Enumeration
Reidemeister-Schreier
Quotient Algorithms
Knuth-Bendix

Matrix groups over finite ~~fields~~ Rings:
Composition tree
Homomorphisms
Finite group calculations

Local/Global:
Congruence images and their structure

rithmic Tools

Weigel
1995,96



Matthews,
Vaserstein,
Weisfeiler
1984

groups:

Reidemeister-Schreier
Quotient Algorithms
Knuth-Bendix

Matrix groups over finite ~~fields~~ Rings:

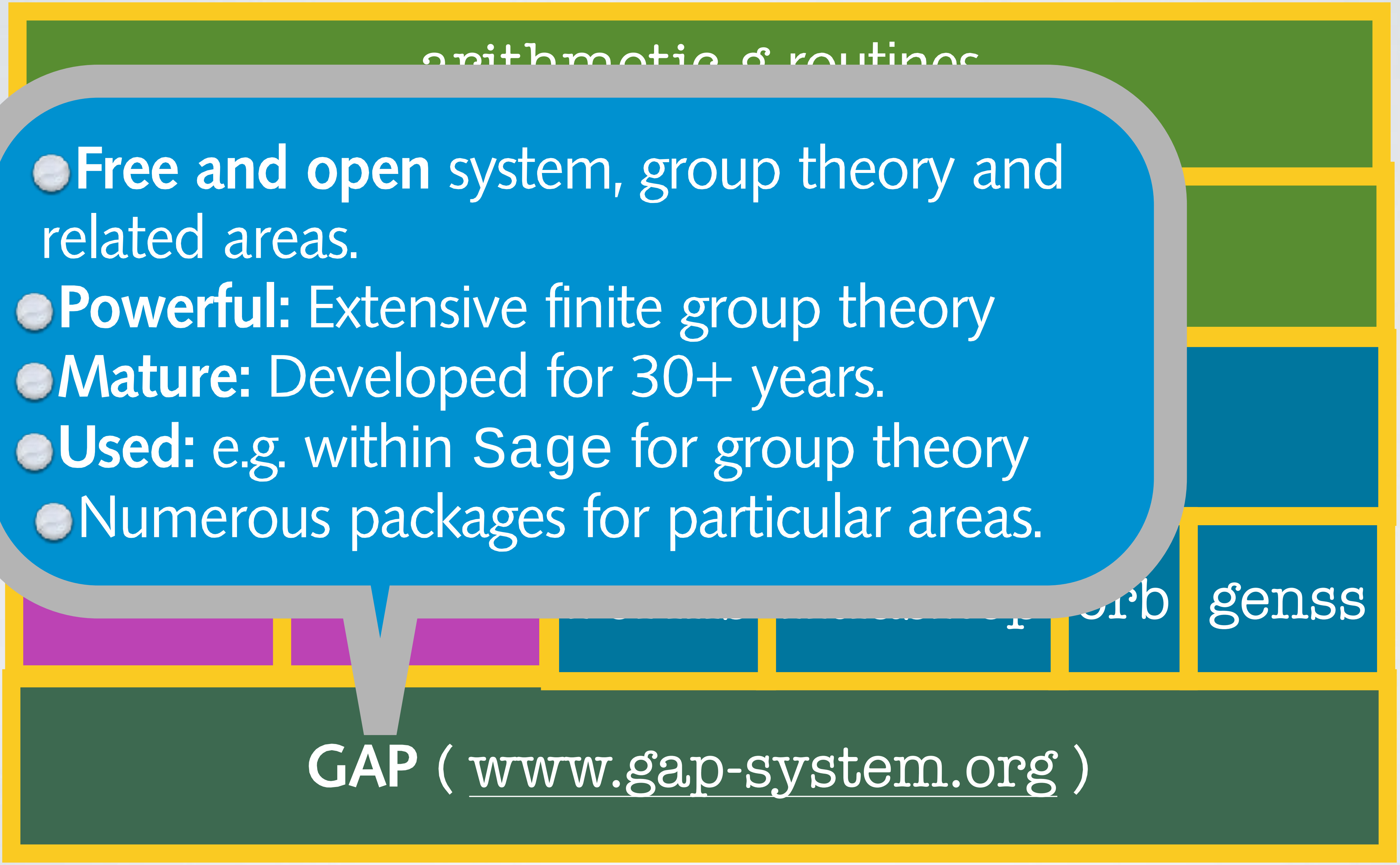
Composition tree
Homomorphisms
Finite group calculations

CSP (e.g. dim ≥ 3): Finite index ⇒ Dense ⇔
$\varphi_p(G) = \varphi_p(\Gamma)$ modulo almost all primes

Local/Global:

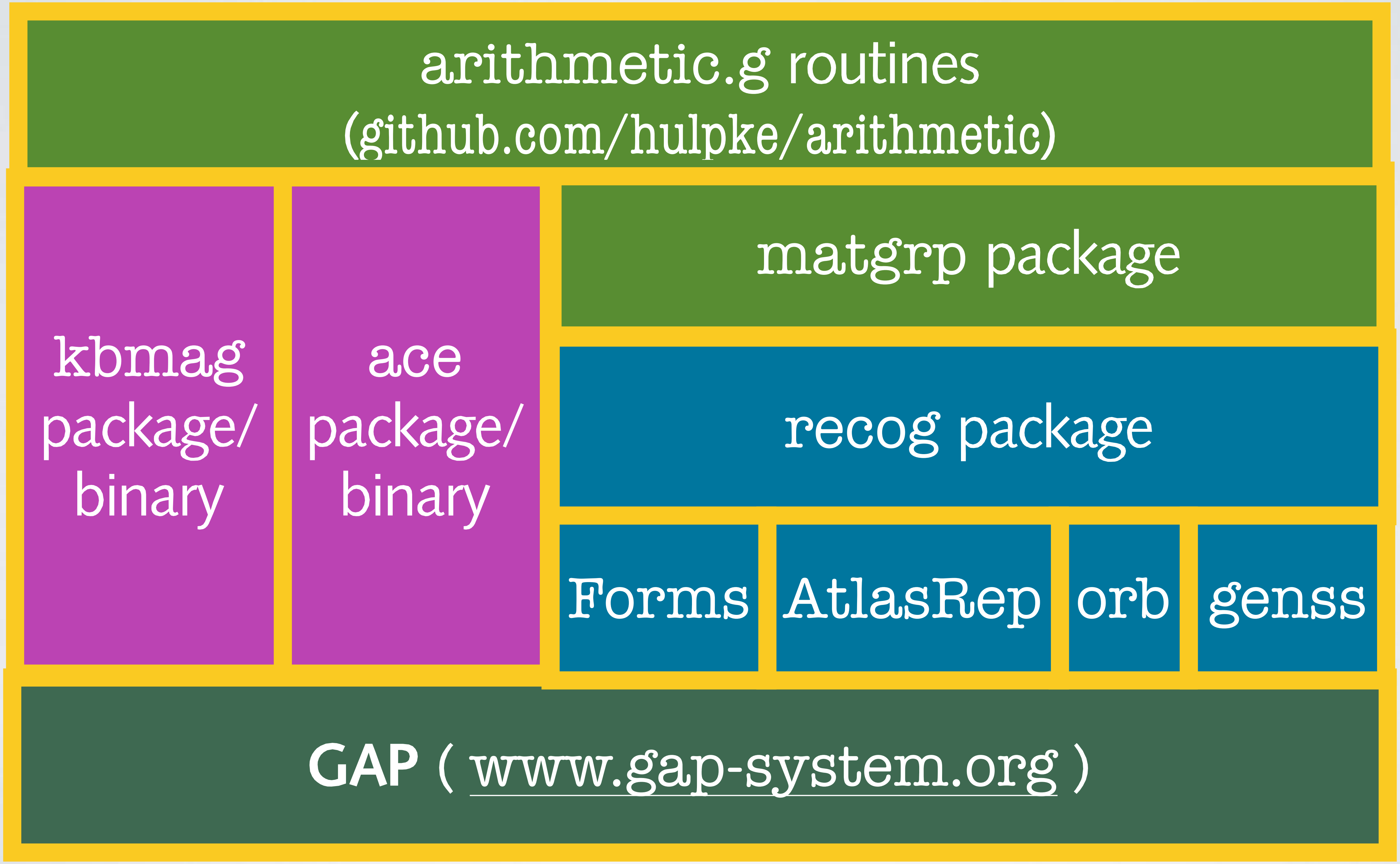
Congruence images and their structure

Software Stack Used



- **Free and open** system, group theory and related areas.
- **Powerful:** Extensive finite group theory
- **Mature:** Developed for 30+ years.
- **Used:** e.g. within Sage for group theory
- Numerous packages for particular areas.

Software Stack Used



arithmetic.g routines
(github.com/hulpke/arithmetic)

kbmag package/ binary

ace package/ binary

matgrp package

recog package

Forms

AtlasRep

orb

genss

GAP (www.gap-system.org)

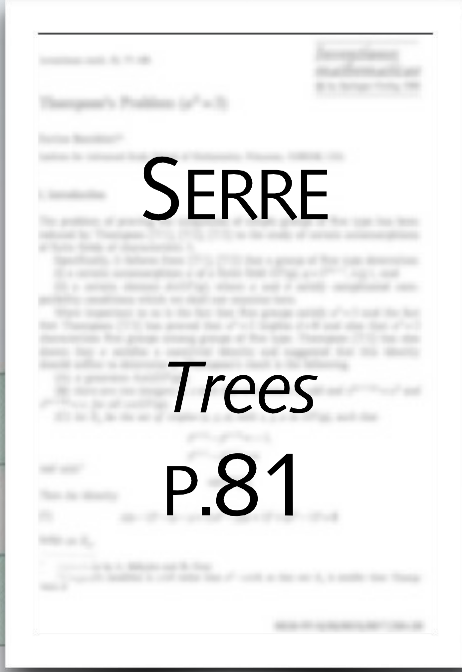
```
            GAP 4.12dev built on 2021-06-07
  GAP       https://www.gap-system.org
            Architecture: x86_64-apple-darwin20.5.0
gap> Read("arithmetic.g");
Loading 'Forms' 1.2.5
Loading  orb 4.8.3 (Methods to enumerate orbits)
Loading  genss 1.6.5 (Generic Schreier-Sims)
Loading  recog 1.3.2dev (Group recognition methods)
Matrix Group Interface routines
Arithmetic group routines, Version 1.11, Alexander Hulpke
gap> psi:=SLNZFP(3);    # also SPNZFP
[ t12, t13, t21, t23, t31, t32 ] ->
[ [ [ 1, 1, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ],[…] ]
gap> F:=Source(psi);
<fp group on [ t12, t13, t21, t23, t31, t32 ]>
gap> Range(psi);
<matrix group with 6 generators>
gap> Image(psi,PseudoRandom(F:radius:=100));
[[1203,-1886,1904],[1381,-2181,2192],[-2498,3935,-3961]]
gap> PseudoRandom(Range(psi)); #forced short to 20 minutes
[ [ <integer 144...984 (161571960 digits)>,[…] ]
```

Constructing $SL_2(\mathbb{Z}[1/7])$

Following SERRE (choose $p = 7$).

Take $SL_2(\mathbb{Z}) = \langle a, b \mid a^4, a^2 = (ab)^3 \rangle$

```
gap> f:=FreeGroup("a","b");;
gap> g:=f/ParseRelators(f,"a4,a2=(ab)3");
<fp group on the generators [ a, b ]>
```



SERRE

Trees

P.81

Two matrix representations: $\alpha\colon a, b \mapsto \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and

$\beta\colon a, b \mapsto \begin{pmatrix} 0 & 1/7 \\ -7 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1/7 \\ 0 & 1 \end{pmatrix}$.

```
gap> mats:=[[[0,1],[-1,0]],[[1,0],[1,1]]];;
gap> alpha:=GroupHomom...ByImages(g,Group(mats),GeneratorsOfGroup(g),mats);
[ a, b ] -> [[[0,1], [-1,0] ], [ [ 1, 0 ], [ 1, 1 ] ] ]
gap> new:=[[[0,1/7], [-7,0] ], [ [1,-1/7], [0,1] ] ];;
gap> beta:=GroupHomomorphismByImages(g,Group(new),GeneratorsOfGroup(g),new);
[ a, b ] -> [[[0,1/7],[-7,0]], [[1,-1/7],[0,1]]]
```

Nonprincipal congruence subgroup $\begin{pmatrix} * & * \\ 7* & * \end{pmatrix}$ modulo 7:

```
gap> one:=One(GF(7));;mare:=mats*one;;
gap> red:=GroupHomomorphismByImages(g,Group(mare),GeneratorsOfGroup(g),mare);
[ a, b ] -> [ [ [0*Z(7),Z(7)^0 ],[Z(7)^3,0*Z(7)]],[…]  ]
gap> sub:=Stabilizer(Range(red),[0,1]*one,OnLines);
Group([ [ Z(7)^2, Z(7)^3 ], [ 0*Z(7), Z(7)^4 ] ], […])
gap> sub:=PreImage(red,sub);
Group(<fp, no generators known>)
gap> gens:=GeneratorsOfGroup(sub);
[a^2,a*b/a,b^2/a^2/b^2,b^3/a/b^2,(a^-2)^(b^2),b^-3/a*b^2]
gap> gens:=gens{[1,2,4,6]};;
gap> suma:=List(gens,x->Image(beta,x));
[[[-1,0],[0,-1]],[[1,0],[7,1]],[[-3,-1],[7,2]],[…]]
```

Factor generators (done on the side — see later)

```
gap> wrd:=[a^2,b^7,(a^-1*b)^(b^2),(a^-1*b^-1)^(b^-2)];;
gap> List(wrd,x->ImagesRepresentative(alpha,x))=suma;
true # verified words
```

Redundancy amongst generators:

```
gap> suma:=List(gens,x->ImagesRepresentative(beta,x));
[ [ [ -1, 0 ], [ 0, -1 ] ], [ [ 1, 0 ], [ 7, 1 ] ],
  [ [ -1, 0 ], [ 0, -1 ] ], [ [ -3, -1 ], [ 7, 2 ] ],
  [ [ -1, 0 ], [ 0, -1 ] ], [ [ 3, -1 ], [ 7, -2 ] ] ]
```

```
gap> sub:=Stabilizer(Range(red),[0,1]*One,OnLines);
Group([ [ Z(7)^2, Z(7)^3 ], [ 0*Z(7), Z(7)^4 ] ], […])
gap> sub:=PreImage(red,sub);
Group(<fp, no generators known>)
gap> gens:=GeneratorsOfGroup(sub);
[a^2,a*b/a,b^2/a^2/b^2,b^3/a/b^2,(a^-2)^(b^2),b^-3/a*b^2]
gap> gens:=gens{[1,2,4,6]};;
gap> suma:=List(gens,x->Image(beta,x));
[[[-1,0],[0,-1]],[[1,0],[7,1]],[[-3,-1],[7,2]],[…]]
```

Factor generators (done on the side — see later)

```
gap> wrd:=[a^2,b^7,(a^-1*b)^(b^2),(a^-1*b^-1)^(b^-2)];;
gap> List(wrd,x->ImagesRepresentative(alpha,x))=suma;
true # verified words
```

Nonprincipal congruence subgroup $\begin{pmatrix} * & * \\ 7* & * \end{pmatrix}$ modulo 7:

```
gap> one:=One(GF(7));;mare:=mats*one;;
gap> red:=GroupHomomorphismByImages(g,Group(mare),GeneratorsOfGroup(g),mare);
[ a, b ] -> [ [ [0*Z(7),Z(7)^0 ],[Z(7)^3,0*Z(7)]],[…]  ]
gap> sub:=Stabilizer(Range(red),[0,1]*one,OnLines);
Group([ [ Z(7)^2, Z(7)^3 ], [ 0*Z(7), Z(7)^4 ] ], […])
gap> sub:=PreImage(red,sub);
Group(<fp, no generators known>)
gap> gens:=GeneratorsOfGroup(sub);
[a^2,a*b/a,b^2/a^2/b^2,b^3/a/b^2,(a^-2)^(b^2),b^-3/a*b^2]
gap> gens:=gens{[1,2,4,6]};;
gap> suma:=List(gens,x->Image(beta,x));
[[[-1,0],[0,-1]],[[1,0],[7,1]],[[-3,-1],[7,2]],[…]]
```

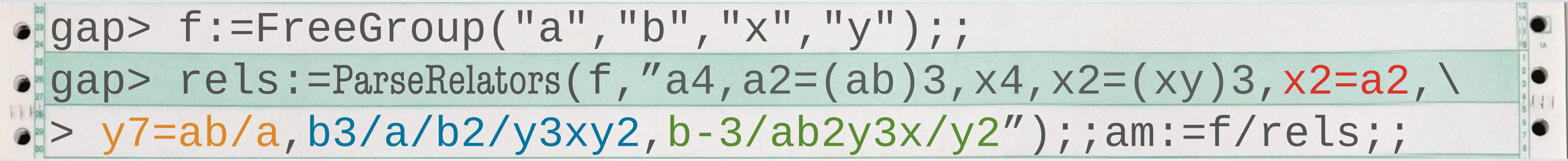
Factor generators (done on the side — see later)

```
gap> wrd:=[a^2,b^7,(a^-1*b)^(b^2),(a^-1*b^-1)^(b^-2)];;
gap> List(wrd,x->ImagesRepresentative(alpha,x))=suma;
true # verified words
```

$$a^2, ab/a, b^3/a/b^2, b^{-3}a^{-1}b^2$$

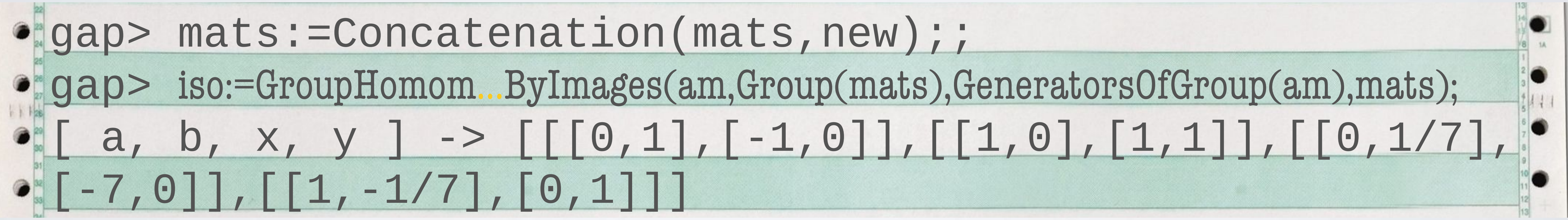
$$a^2, b^7, (a^{-1}b)^{b^2}, (a^{-1}b^{-1})^{b^{-2}}$$

Amalgamate the subgroups through generators



```
gap> f:=FreeGroup("a","b","x","y");;
gap> rels:=ParseRelators(f,”a4,a2=(ab)3,x4,x2=(xy)3,x2=a2,\
> y7=ab/a,b3/a/b2/y3xy2,b-3/ab2y3x/y2”);;am:=f/rels;;
```

Combine the matrix representations:



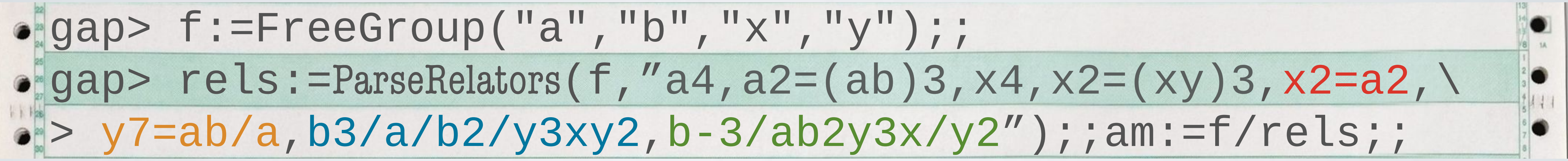
```
gap> mats:=Concatenation(mats,new);;
gap> iso:=GroupHomom…ByImages(am,Group(mats),GeneratorsOfGroup(am),mats);
[ a, b, x, y ] -> [[[0,1],[-1,0]],[[1,0],[1,1]],[[0,1/7],
[-7,0]],[[1,-1/7],[0,1]]]
```

Since we did not add an NC, this verified that the map is a homomorphism.

$$a^2, ab/a, b^3/a/b^2, b^{-3}a^{-1}b^2$$

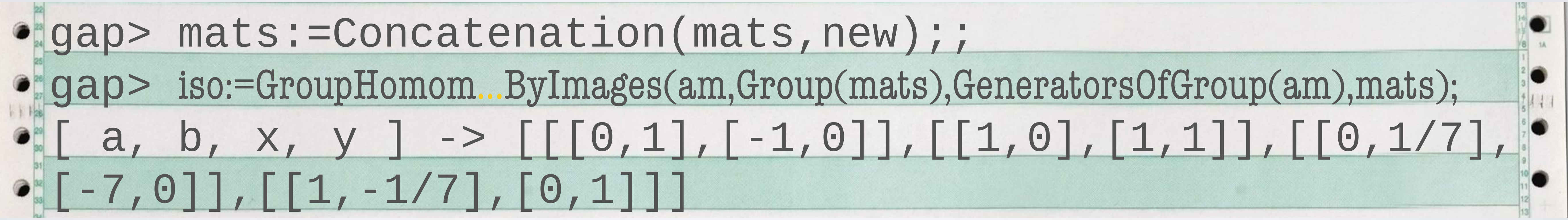
$$x^2, y^7, (x^{-1}y)^{y^2}, (x^{-1}y^{-1})^{y^{-2}}$$

Amalgamate the subgroups through generators



```
gap> f:=FreeGroup("a","b","x","y");;
gap> rels:=ParseRelators(f,”a4,a2=(ab)3,x4,x2=(xy)3,x2=a2,\
> y7=ab/a,b3/a/b2/y3xy2,b-3/ab2y3x/y2”);;am:=f/rels;;
```

Combine the matrix representations:



```
gap> mats:=Concatenation(mats,new);;
gap> iso:=GroupHomom...ByImages(am,Group(mats),GeneratorsOfGroup(am),mats);
[ a, b, x, y ] -> [[[0,1],[-1,0]],[[1,0],[1,1]],[[0,1/7],
[-7,0]],[[1,-1/7],[0,1]]]
```

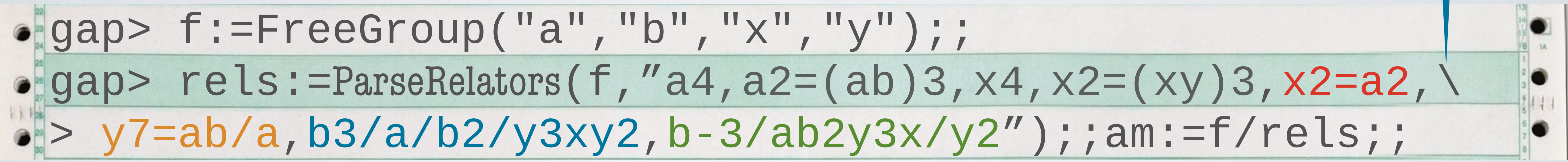
Since we did not add an NC, this verified that the map is a homomorphism.

$$a^2, ab/a, b^3/a/b^2, b^{-3}a^{-1}b^2$$

$$x^2, y^7, (x^{-1}y)^{y^2}, (x^{-1}y^{-1})^{y^{-2}}$$

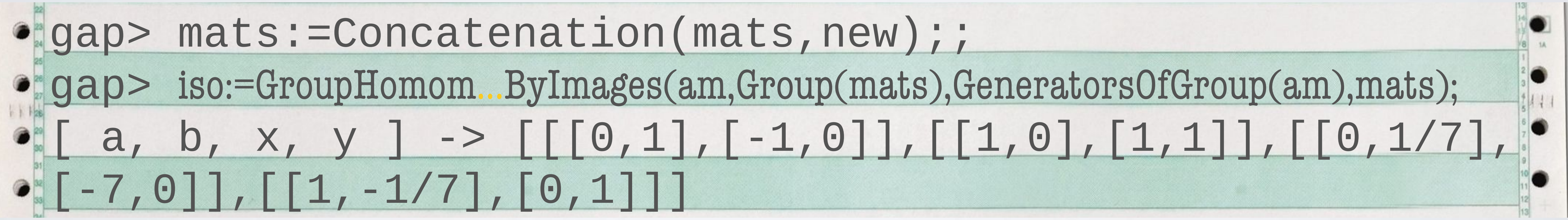
String goes over more than one line

Amalgamate the subgroups through generators



```
gap> f:=FreeGroup("a","b","x","y");;
gap> rels:=ParseRelators(f,”a4,a2=(ab)3,x4,x2=(xy)3,x2=a2,\
> y7=ab/a,b3/a/b2/y3xy2,b-3/ab2y3x/y2”);;am:=f/rels;;
```

Combine the matrix representations:



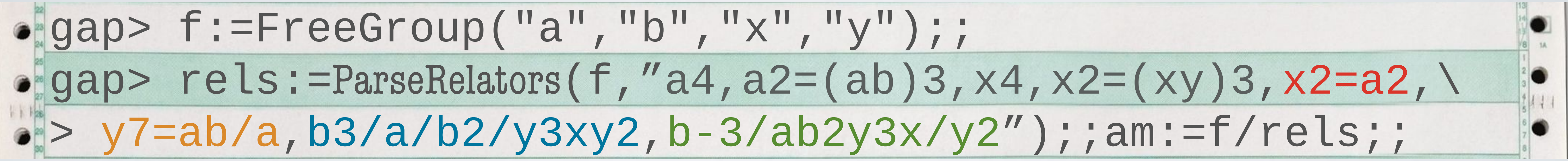
```
gap> mats:=Concatenation(mats,new);;
gap> iso:=GroupHomom...ByImages(am,Group(mats),GeneratorsOfGroup(am),mats);
[ a, b, x, y ] -> [[[0,1],[-1,0]],[[1,0],[1,1]],[[0,1/7],
[-7,0]],[[1,-1/7],[0,1]]]
```

Since we did not add an NC, this verified that the map is a homomorphism.

$$a^2, ab/a, b^3/a/b^2, b^{-3}a^{-1}b^2$$

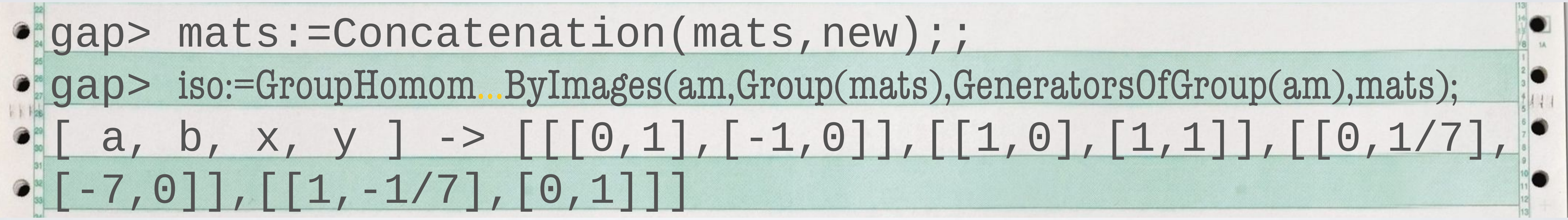
$$x^2, y^7, (x^{-1}y)^{y^2}, (x^{-1}y^{-1})^{y^{-2}}$$

Amalgamate the subgroups through generators



```
gap> f:=FreeGroup("a","b","x","y");;
gap> rels:=ParseRelators(f,”a4,a2=(ab)3,x4,x2=(xy)3,x2=a2,\
> y7=ab/a,b3/a/b2/y3xy2,b-3/ab2y3x/y2”);;am:=f/rels;;
```

Combine the matrix representations:



```
gap> mats:=Concatenation(mats,new);;
gap> iso:=GroupHomom...ByImages(am,Group(mats),GeneratorsOfGroup(am),mats);
[ a, b, x, y ] -> [[[0,1],[-1,0]],[[1,0],[1,1]],[[0,1/7],
[-7,0]],[[1,-1/7],[0,1]]]
```

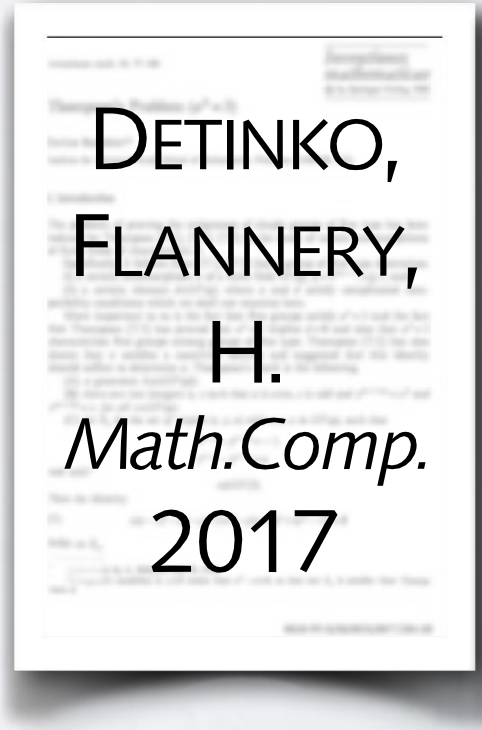
Since we did not add an NC, this verified that the map is a homomorphism.

Index In Congruence Images

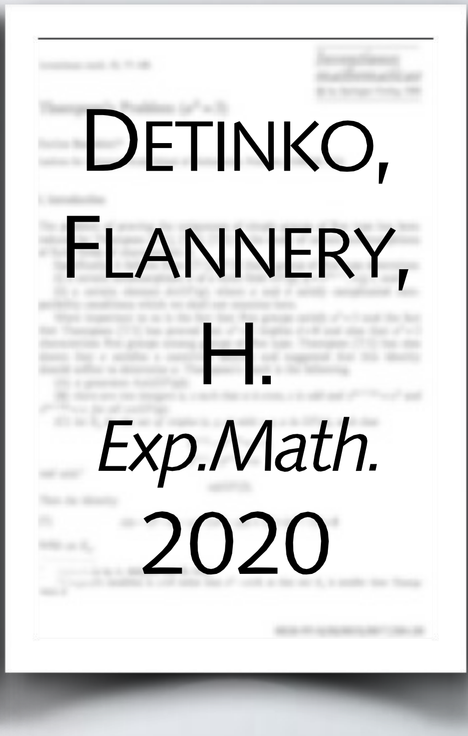
Congruence images are approximations. There is a largest principal congruence subgroup $\Gamma_l \leq \Gamma$ of level l, such that $\langle \Gamma_l, G \rangle \leq \langle \Gamma_k, G \rangle$ for any k. Find this level l, and index $[\Gamma{:}\langle \Gamma_l, G \rangle] = [\varphi_l(\Gamma){:}\varphi_l(G)]$.

Two part problem:

- Find primes dividing l
- Find l from primes.



DETINKO,
FLANNERY,
H.
Math.Comp.
2017



DETINKO,
FLANNERY,
H.
Exp.Math.
2020

Structure Of Congruence Image

Image $A_l = \varphi_l(\Gamma)$ is classical group (SL/Sp) over $Z_l = \mathbb{Z}/l\mathbb{Z}$.

A_l subdirect product of A_{p^e} for prime powers p^e in l.

A_{p^e} is iterated extension of adjoint module (Lie sense) by group A_p over prime field.

Generically, these extensions A_{p^e} are not supplemented, thus if $[A_{m\cdot p} : \varphi_{m\cdot p}(G)] = [A_m : \varphi_m(G)]$, no further factor of p in l. Thus find maximal prime powers.

Structure Of Congruence Image

Image $A_l = \varphi_l(\Gamma)$ is classical group (SL/Sp) o

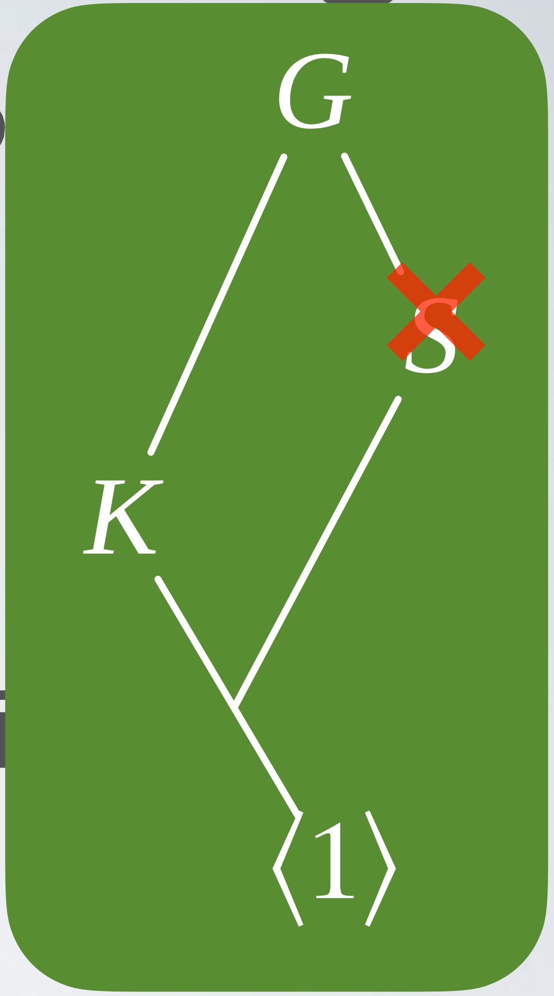
$Z_l = \mathbb{Z}/l\mathbb{Z}$.

A_l subdire e powers

A_{p^e} is itera nodule (Li

by group A_p

Exceptions for small primes (2,3) and small dimensions (≤ 3), resolved by starting with prime power p^2.



Generically, these extensions A_{p^e} are not supplemented, thus if $[A_{m \cdot p} : \varphi_{m \cdot p}(G)] = [A_m : \varphi_m(G)]$, no further factor of p in l. Thus find maximal prime powers.

Finding Primes

Same argument, primes dividing l are (disclaimer for $p = 2,3$) those p, such that $\varphi_p(G) \neq A_p$.

Criteria: a) Given transvection stays transvection

b) Adjoint repres. of G mod p abs. red.

Theorem: An (irreducible) representation $\alpha : G \to \mathrm{GL}_k(\mathbb{Z})$ is absolutely irreducible modulo every prime that does not divide the discriminant of the $\mathbb{Z}$-lattice spanned by $\alpha(G)$.

Method: Approximate this lattice until full rank.

Problem: Dimension k^2. (Too) costly for adj. rep.

Finding Primes

Same argument, primes dividing l are (disclaimer for $p = 2,3$) those p, such that $\varphi_p(G) \neq A_p$.

Criteria: a) Given transvection stays transvection

b) Adjoint repres. of G abs. red. mod p

Theorem: An (irreducible) representation $\alpha \colon G \to \mathrm{GL}_k(\mathbb{Z})$ is absolutely irreducible modulo every prime that does not divide the discriminant of the $\mathbb{Z}$-lattice spanned by $\alpha(G)$.

Method: Approximate this lattice until full rank.

Problem: Dimension k^2. (Too) costly for adj. rep.

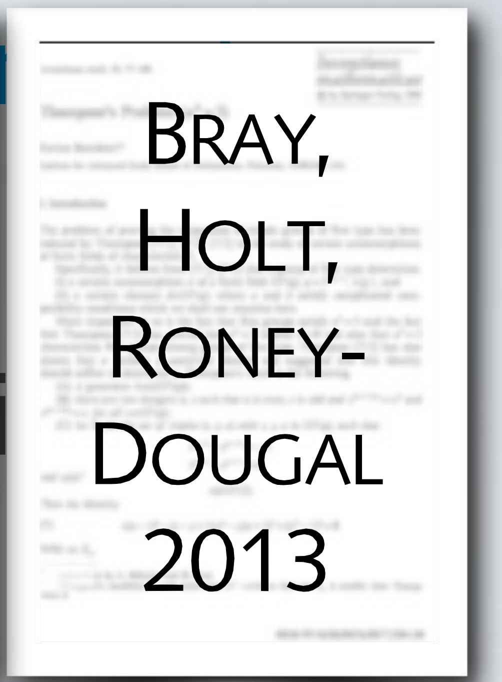
Finding Primes

Same argument, primes dividing l a

$p = 2{,}3$) those p, such that $\varphi_p(G)$

Criteria: a) Given transvection stays

b) Adjoint repres. of G abs. red. m



To find in practice:

- $\varphi_p(G)$ is in maximal subgroup of A_p.
- Aschbacher classes for maximal subgroups — stabilizers of geometric structures, order bounds.
- Translate stabilization to calculation on entries of element(s) in G. Test prime divisors of value.

Finding Primes

S es dividing l are (disclaimer for

hat $\varphi_p(G) \neq A_p$.

Crit ansvection stays transvection

b) Adjoint re es. of G abs. red. mod p

To find in prac ce:

- $\varphi_p(G)$ is in m ximal subgroup of A_p.
- Aschbacher cla es for maximal subgroups — stabilizers of geo etric structures, order bounds.
- Translate stabilization to calculation on entries of element(s) in G. Test prime divisors of value.

Any such prime is candidate only — test whether $\varphi_p(G) \neq A_p$

For Example

Convention: $a \in A_p$. $g \in G$ (pseudo-)random.

Order Bound: $|a| \leq b$ implies exponent $a^e = 1$. For $|g| = \infty$, consider lcm of entries of $g^e - 1$.

Tensor Product: If $a = a_1 \otimes a_2$, then coefficients of $\chi_a(x) = \prod_{i,j} (x - \alpha_i \beta_j)$ satisfy syzygy condition.

Reducible (not absolutely)**:** Run irreducibility test (MeatAxe) generically over $\mathbb{Z}$. Collect primes modulo which reduction fails.

+ further cases …

For Example

Hard (?) commutative algebra problem

Convention: $a \in A_p$. $g \in G$ (pseudo-)random.

Order Bound: $|a| \leq b$ implies exponent $a^e = 1$. For $|g| = \infty$, consider lcm of entries of $g^e - 1$

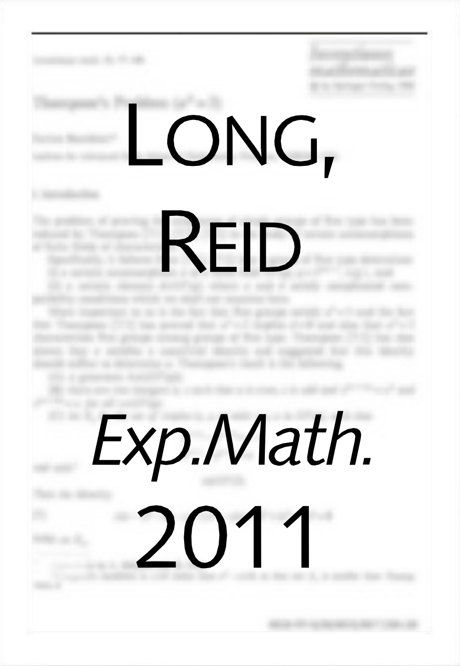
Tensor Product: If $a = a_1 \otimes a_2$, then coefficients of $\chi_a(x) = \prod_{i,j} (x - \alpha_i \beta_j)$ satisfy syzygy condition.

Reducible (not absolutely): Run irreducibility test (MeatAxe) generically over $\mathbb{Z}$. Collect primes modulo which reduction fails.

+ further cases …


GAP RD
STOP

```
gap> g:=BetaT(27);
<matrix group with 3 generators>
gap> pr:=PrimesNonSurjective(g);
#I  Absolute irreducibility - found: [ 17, 23,
#I  Element Order - found: [ 3 ] new:[  ]
#I  Similarity - found: [ 3, 17, 23, 151 ] new:
#I  Monomial - found: [ 3 ] new:[  ]
#I  Solvable - found: [ 3 ] new:[  ]
[ 3, 17, 23, 151 ]
gap> MaxPCSPrimes(g,pr);
#I  Try 3^1, Try 3^2, […] Try 3^10
#I  Try 17^1, Try 17^2
#I  Try 23^1, Try 23^2
#I  Try 151^1, Try 151^2
#I  Try 1162104003 3
#I  Try 1162104003 17
#I  Try 1162104003 23
#I  Try 1162104003 151
#I  Try extra 2
#I  Level = 1162104003 =3^9*17*23*151
[ 1162104003, 9854089154390043625196124257904 ]
```



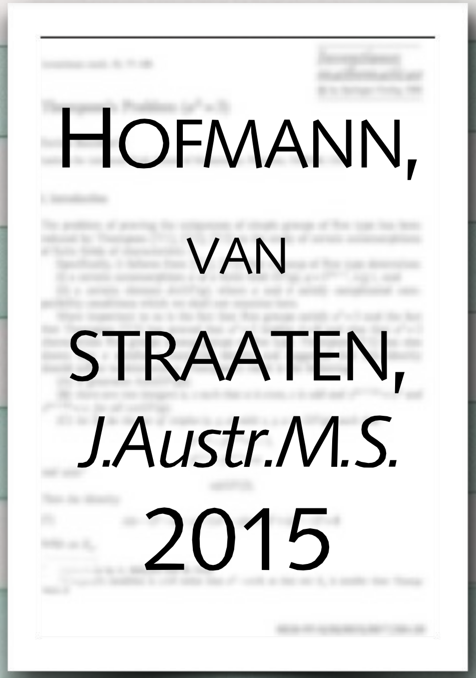
LONG, REID

Exp.Math. 2011

Examples chosen to illustrate, even if results known.

```
gap> g:=BetaT(27);
<matrix group with 3 generators>
gap> pr:=PrimesNonSurjective(g);
#I  Absolute irreducibility - found: [ 17, 23, 151 ]
#I  Element Order - found: [ 3 ] new:[  ]
#I  Similarity - found: [ 3, 17, 23, 151 ] new:[  ]
#I  Monomial - found: [ 3 ] new:[  ]
#I  Solvable - found: [ 3 ] new:[  ]
[ 3, 17, 23, 151 ]
gap> MaxPCSPrimes(g,pr);
#I  Try 3^1, Try 3^2, […] Try 3^10
#I  Try 17^1, Try 17^2
#I  Try 23^1, Try 23^2
#I  Try 151^1, Try 151^2
#I  Try 1162104003 3
#I  Try 1162104003 17
#I  Try 1162104003 23
#I  Try 1162104003 151
#I  Try extra 2
#I  Level = 1162104003 =3^9*17*23*151
[ 1162104003, 9854089154390043625196124257904 ]
```

```
gap> g:=HofmannStraatenExample(3,4);  # Calabi-Yau Threefolds
<matrix group with 2 generators>
gap> RankMat(g.2-g.2^0);
1
gap> PrimesForDense(g,g.2,SP); # use transvection
[ 2, 3 ]
gap> MaxPCSPrimes(g,pr,SP);
#I  Try 2^1, 2^2, 2^3
#I  Try 3^1, 3^2, 3^3
#I  Try 36 2
#I  Try 36 3
#I  Level = 36 = 2^2*3^2
#I  Index = 3110400 =2^9*3^5*5^2
[ 36, 3110400 ]
```



HOFMANN,
VAN
STRAATEN,
J.Austr.M.S.
2015

```
ga                           mple(3,4);  # Calabi-Yau Threefolds
<m                           rators>
gap>                         ,
1
gap> Primes    Dense(g,g.2,SP); # use transvection
[ 2, 3 ]
gap> MaxPCS   imes(g,pr,SP);
#I  Try 2^1   2^2, 2^3
#I  Try 3^1   3^2, 3^3
#I  Try 36  2
#I  Try 36  3
#I  Level  = 36 = 2^2*3^2
#I  Index  = 3110400 =2^9*3^5*5^2
[ 36, 3110400 ]
```

This
ONLY is the index of
the smallest congruence
subgroup above, not
of G

Verifying The Index

$$F \xrightarrow{\psi} \Gamma \xrightarrow{\varphi_m} A_m$$

Unless we know the subgroup to be a congruence subgroup , the maximal index in a congruence image is only the index of the smallest congruence subgroup above, and will never show infinite index.

Thus consider problem in finitely presented pre-image under ψ :

- Pre-image of matrix is word in generators
- Hope fp-group methods finish

Finding Words

$F \xrightarrow{\psi} \Gamma \xrightarrow{\varphi_m} A_m$

To find pre-image under ψ, need to express matrix in Γ as product of elementary generators:

Try possible words of short length: Limited scope

Word in A_m: Deterministic, but might need huge m.

Hermite NF: Only for SL, Long Words: $\begin{pmatrix} 1 & 34 \\ 1000 & 21 \end{pmatrix}$, We care not about Normal Form but transform matrix.

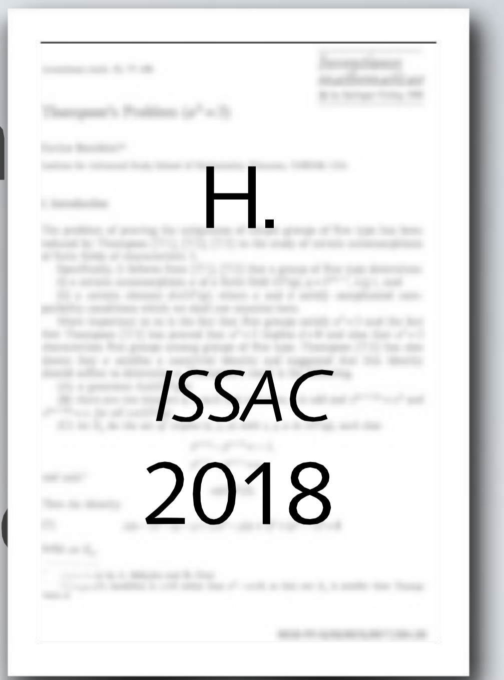
Ultimately Use: Norm-based reduction (row+column) heuristics. Produces reasonable words for SL and Sp.

Finding Words

$$F \xrightarrow{\psi} \Gamma \xrightarrow{\varphi_m} A_m$$

To find pre-image under ψ, need to express m

Γ as product of elementary generators:



H.

ISSAC

2018

Try possible words of short length: Limited sc

Word in A_m: Deterministic, but might need huge m.

Hermite NF: Only for SL, Long Words: $\begin{pmatrix} 1 & 34 \\ 1000 & 21 \end{pmatrix}$, We care not about Normal Form but transform matrix.

Ultimately Use: Norm-based reduction (row+column) heuristics. Produces reasonable words for SL and Sp.

```
gap> g:=HofmannStraatenExample(3,4);;  # as before — 36, 3110400
gap> psi:=SPNZFP(4);;F:=Source(psi);
<fp group on the generators [ Y1, Y2, U1, U2, Z1 ]>
gap> w:=List(GeneratorsOfGroup(g),
>     x->PreImagesRepresentative(psi,x));
[ U1*(U2^-1*U1*U2^-1)^2*Y1^-1*Y2^-1*U2^-1*Y2^-1*Z1*U2*Y2, Y2^-1 ]
gap> S:=Subgroup(F,w);;
gap> Index(F,S);
3110400    # have proven the finite index, 5 minutes

gap> g:=HofmannStraatenExample(1,4);;  # known infinite index
gap> pr:=PrimesForDense(g,g.2,SP);
[ 2 ]
gap> MaxPCSPrimes(g,pr,SP);
[ 4, 160 ]  # level 4, index 160
gap> w:=List(GeneratorsOfGroup(g),x->PreImagesRepresentative(psi,x));
[ U2^-3*U1*U2^-1*Y1^-1*Y2^-1*U2^-1*Y2^-1*Z1*U2*Y2, Y2^-1]
gap> Index(F,S);
Error, coset enumeration defined more than 4096000 cosets
```

```
gap> g:=HofmannStraatenExample(3,4);;  # as before — 36, 3110400
gap> psi:=SPNZFR(4);;F:=Source(psi);
<fp gr                    ators [ Y1, Y2, U1, U2, Z1 ]>
gap                       fGroup(g),
>                         tative(psi,x));
[ U                       -1*U2^-1*Y2^-1*Z1*U2*Y2, Y2^-1 ]
gap>
gap> Ind
3110400    # h   e proven the finite index, 5 minutes

gap> g:=Hofma  StraatenExample(1,4);;  # known infinite index
gap> pr:=Prim  ForDense(g,g.2,SP);
[ 2 ]
gap> MaxPCSPri es(g,pr,SP);
[ 4, 160 ]   #  evel 4, index 160
gap> w:=List(Genera orsOfGroup(g),x->PreImagesRepresentative(psi,x));
[ U2^-3*U1*U2^ 1*Y1^-1*Y2^-1*U2^-1*Y2^-1*Z1*U2*Y2, Y2^-1]
gap> Index(F,S);
Error, coset enumeration defined more than 4096000 cosets
```

This is **NOT** a proof that the index is infinite, just failure so far.

```
gap> g:=HofmannStraatenExample(4,4);;  # known finite index
gap> pr:=PrimesForDense(g,g.2,SP);
[2]
gap> MaxPCSPrimes(g,pr,SP);
[ 64, 47185920 ]
gap> w:=List(GeneratorsOfGroup(g),x->PreImagesRepresentative(psi,x));
[ U1^2*(U2^-1*U1*U2^-1)^2*Y1^-1*Y2^-1*U2^-1*Y2^-1*Z1*U2*Y2, Y2^-1 ]
gap> S:=Subgroup(F,w);;
gap> Index(F,S);
Error, coset enumeration defined more than 4096000 cosets
brk> return;
[…]    coset enumeration defined more than 16384000 cosets
gap> LoadPackage("ace");
Loading    ACE (C code by  George Havas, Colin Ramsay)
gap> TCENUM:=ACETCENUM;; #change default TC
```

Could use external binary for more powerful coset enumerator. Note that it requires to set options to use more than basic memory:

`:hard, Wo:=10^8`

```
gap> g:=HofmannStraatenE
gap> pr:=PrimesForDense(
[2]
gap> MaxPCSPrimes(g,pr,
[ 64, 47185920 ]
gap> w:=List(GeneratorsOfGroup(       >PreImagesRepresentative(psi,x));
[ U1^2*(U2^-1*U1*U2^-1)^2*Y1^       Y2^-1*U2^-1*Y2^-1*Z1*U2*Y2, Y2^-1 ]
gap> S:=Subgroup(F,w);
gap> Index(F,S);
Error, coset enumeration defined more than 4096000 cosets
brk> return;
[…]     coset enumeration defined more than 16384000 cosets
gap> LoadPackage("ace");
Loading     ACE (C code by  George Havas, Colin Ramsay)
gap> TCENUM:=ACETCENUM;; #change default TC
```

Use Intermediate Subgroup

Coset enumeration must store row for every coset, and intermediately might need more than index.

5 generators, index 47185920 needs at least 30 GB.

Instead find a subgroup $S \lneqq T \lneqq \Gamma$. Rewrite presentation to T and then (in new presentation) calculate $[T:S] = \dfrac{[\Gamma:S]}{[\Gamma:T]}$.

Price: Generator number of T grows with index, but subsequent Tietze transformations tend to reduce.

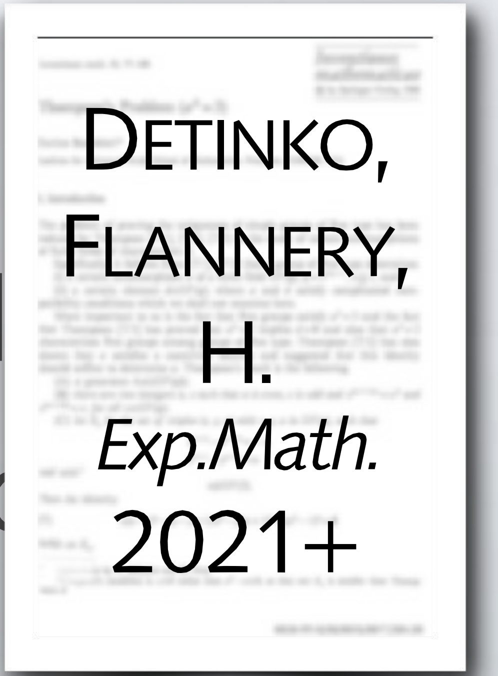
Use Intermediate Subgroup

Coset enumeration must store row for every coset, and intermediately might need more than index.

5 generators, index 47185920 needs at least 30 GB.

Instead find a subgroup $S \lneqq T \lneqq \Gamma$. Rewrite presentation to T and then (in new presentation) calculate $[T:S] = \dfrac{[\Gamma:S]}{[\Gamma:T]}$.

Price: Generator number of T grows with ind
subsequent Tietze transformations tend to red



Detinko, Flannery, H.
Exp.Math.
2021+

Use Intermediate Subgroup

Coset enumeration must store row for every coset, and intermediately mig x.

5 generators, index 47 GB.

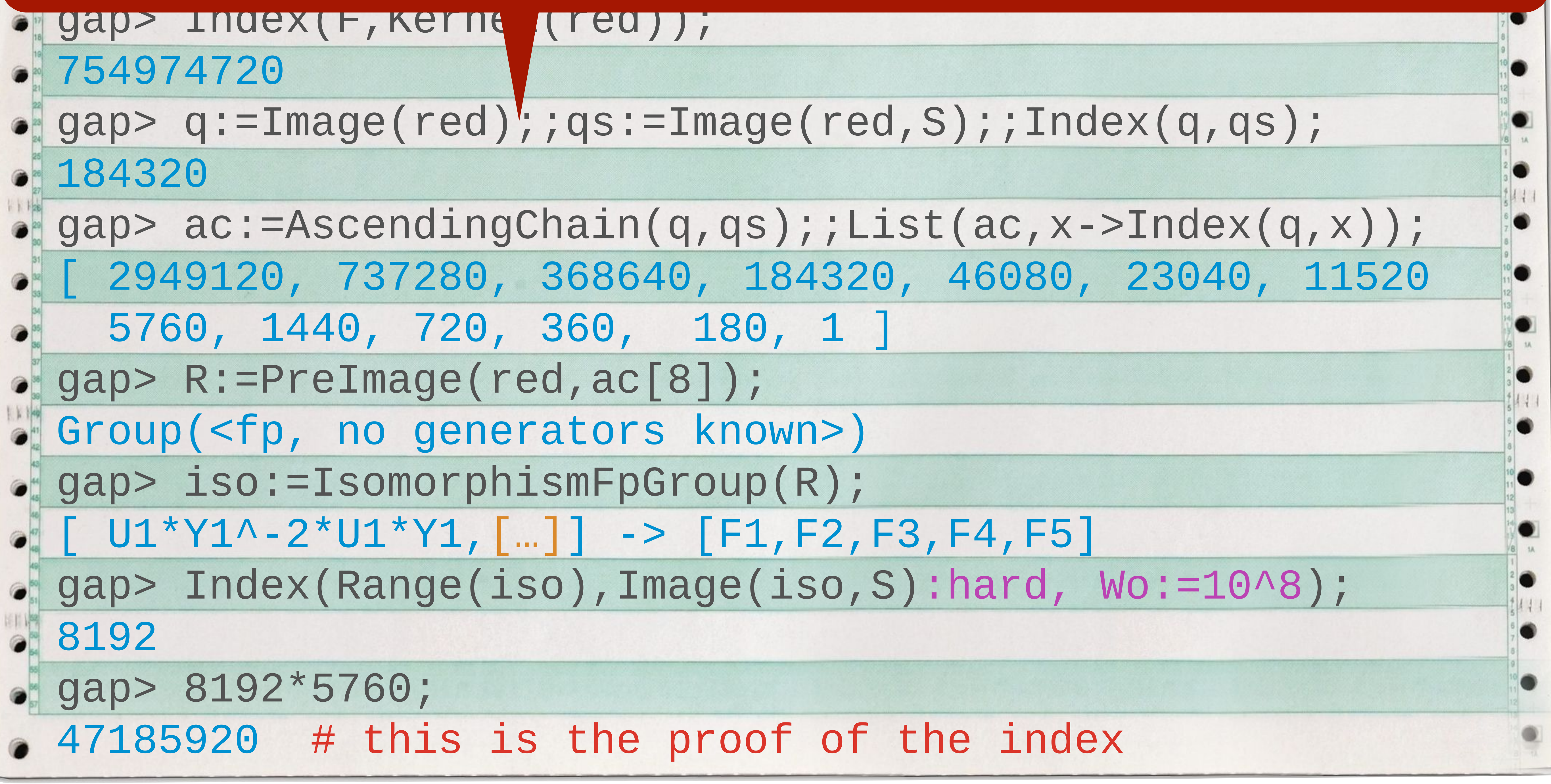
Instead find a subgrou presentation to T an en (in new presentation) calculate $[T : S] = \frac{[\Gamma :}{[\Gamma \quad T]}$.

Price: Generator number of T grows with index, but subsequent Tietze transformations tend to reduce.

Change presentation in sequence of simple steps, trying to delete generators and reduce size.
Heuristic driven.

```
gap> ring:=Integers mod 8;;
gap> red:=MappingGeneratorsImages(psi)[2]*One(ring);
[ [ [ ZmodnZObj( 1, 4 ),[…]]
gap> red:=GroupHomomorphismByImagesNC(F,Group(red),
> MappingGeneratorsImages(psi)[1],red);;
gap> red:=red*IsomorphismPermGroup(Range(red));
gap> Index(F,Kernel(red));
754974720
gap> q:=Image(red);;qs:=Image(red,S);;Index(q,qs);
184320
gap> ac:=AscendingChain(q,qs);;List(ac,x->Index(q,x));
[ 2949120, 737280, 368640, 184320, 46080, 23040, 11520
  5760, 1440, 720, 360,  180, 1 ]
gap> R:=PreImage(red,ac[8]);
Group(<fp, no generators known>)
gap> iso:=IsomorphismFpGroup(R);
[ U1*Y1^-2*U1*Y1,[…]] -> [F1,F2,F3,F4,F5]
gap> Index(Range(iso),Image(iso,S):hard, Wo:=10^8);
8192
gap> 8192*5760;
47185920  # this is the proof of the index
```

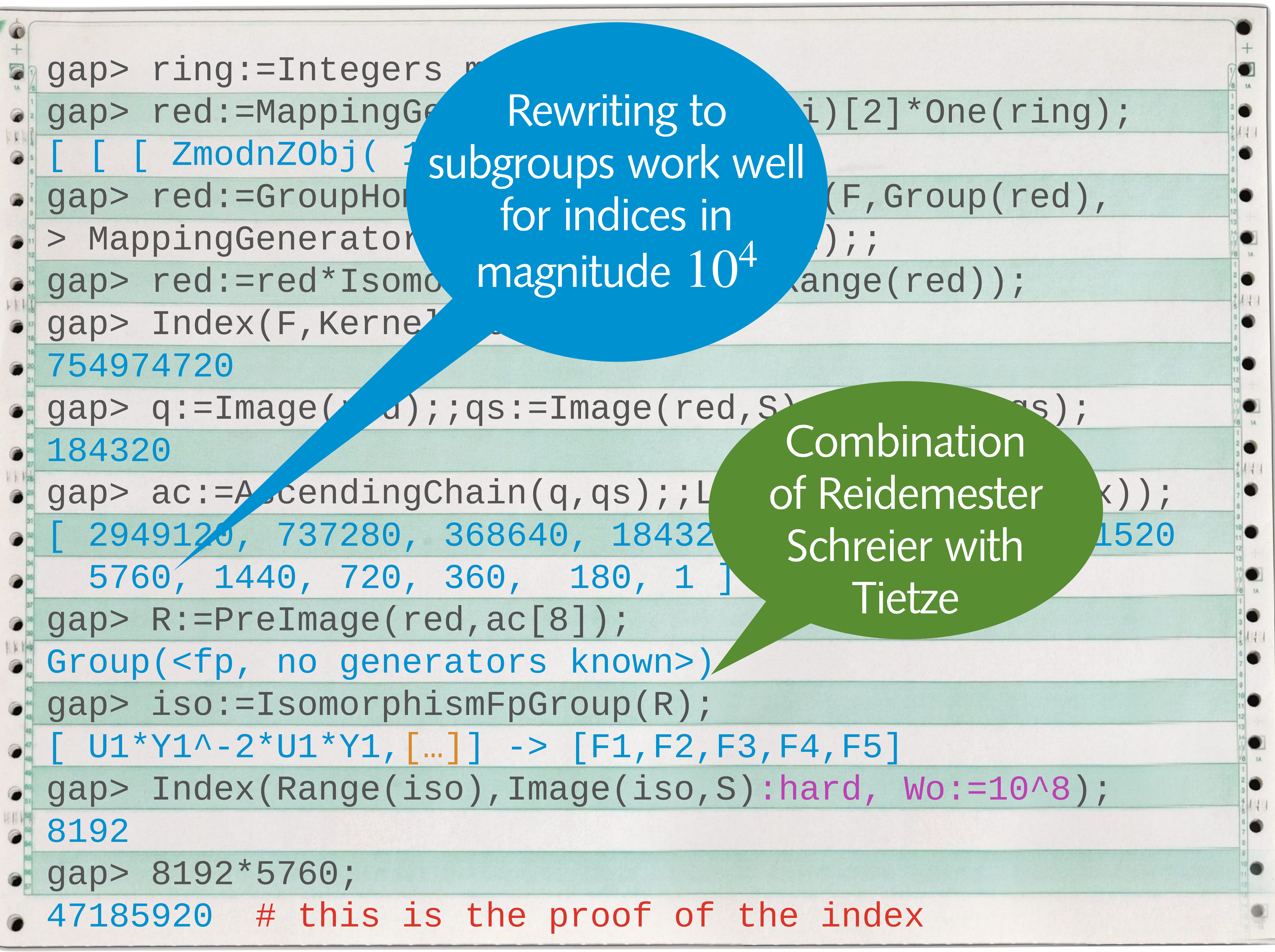
Image is finite group, in which we can do all kinds of calculations. E.g. `IntermediateSubgroups(q,qs);` finds the 2301 subgroups (many b/c of the $p^{k-1} \bmod p^k$ layers) above $\langle \Gamma_8, S \rangle$. Use modulo level (64) to find all finite index subgroups above S.



```
gap> Index(F,Kernel(red));
754974720
gap> q:=Image(red);;qs:=Image(red,S);;Index(q,qs);
184320
gap> ac:=AscendingChain(q,qs);;List(ac,x->Index(q,x));
[ 2949120, 737280, 368640, 184320, 46080, 23040, 11520
  5760, 1440, 720, 360,  180, 1 ]
gap> R:=PreImage(red,ac[8]);
Group(<fp, no generators known>)
gap> iso:=IsomorphismFpGroup(R);
[ U1*Y1^-2*U1*Y1,[…]] -> [F1,F2,F3,F4,F5]
gap> Index(Range(iso),Image(iso,S):hard, Wo:=10^8);
8192
gap> 8192*5760;
47185920  # this is the proof of the index
```

```
gap> ring:=Integers mod 8;;
gap> red:=MappingGeneratorsImages(psi)[2]*One(ring);
[ [ [ ZmodnZObj( 1, 4 ),[…]]
gap> red:=GroupHomomorphismByImagesNC(F,Group(red),
> MappingGeneratorsImages(psi)[1],red);;
gap> red:=red*IsomorphismPermGroup(Range(red));
gap> Index(F,Kernel(red));
754974720
gap> q:=Image(red);;qs:=Image(red,S)                qs);
184320
gap> ac:=AscendingChain(q,qs);;L                    x));
[ 2949120, 737280, 368640, 18432                   1520
  5760, 1440, 720, 360,  180, 1 ]
gap> R:=PreImage(red,ac[8]);
Group(<fp, no generators known>)
gap> iso:=IsomorphismFpGroup(R);
[ U1*Y1^-2*U1*Y1,[…]] -> [F1,F2,F3,F4,F5]
gap> Index(Range(iso),Image(iso,S):hard, Wo:=10^8);
8192
gap> 8192*5760;
47185920  # this is the proof of the index
```

Combination of Reidemester Schreier with Tietze



```
gap> ring:=Integers m
gap> red:=MappingGe                            i)[2]*One(ring);
[ [ [ ZmodnZObj( 1
gap> red:=GroupHo                              (F,Group(red),
> MappingGenerator                             ));;
gap> red:=red*Isomo                            ange(red));
gap> Index(F,Kernel
754974720
gap> q:=Image(    d);;qs:=Image(red,S)              gs);
184320
gap> ac:=AscendingChain(q,qs);;L                    x));
[ 2949120, 737280, 368640, 18432                 1520
   5760, 1440, 720, 360,  180, 1 ]
gap> R:=PreImage(red,ac[8]);
Group(<fp, no generators known>)
gap> iso:=IsomorphismFpGroup(R);
[ U1*Y1^-2*U1*Y1,[…]] -> [F1,F2,F3,F4,F5]
gap> Index(Range(iso),Image(iso,S):hard, Wo:=10^8);
8192
gap> 8192*5760;
47185920  # this is the proof of the index
```

Rewriting to subgroups work well for indices in magnitude 10^4

Combination of Reidemester Schreier with Tietze

Proving Infinite Index

Some attempts of proving infinite index:

- Find an automatic structure that proves infinite index. (Requires suitable embedding.)
  Normal forms for subgroup elements might show that there are infinitely many cosets.
- Show that one can find an infinite index in a suitable quotient image of normal subgroup. (Applicable, if CSP does not hold.)

Proving

The
impossibility results for
algorithms for FPGroups
mean it can never be more
than attempts.

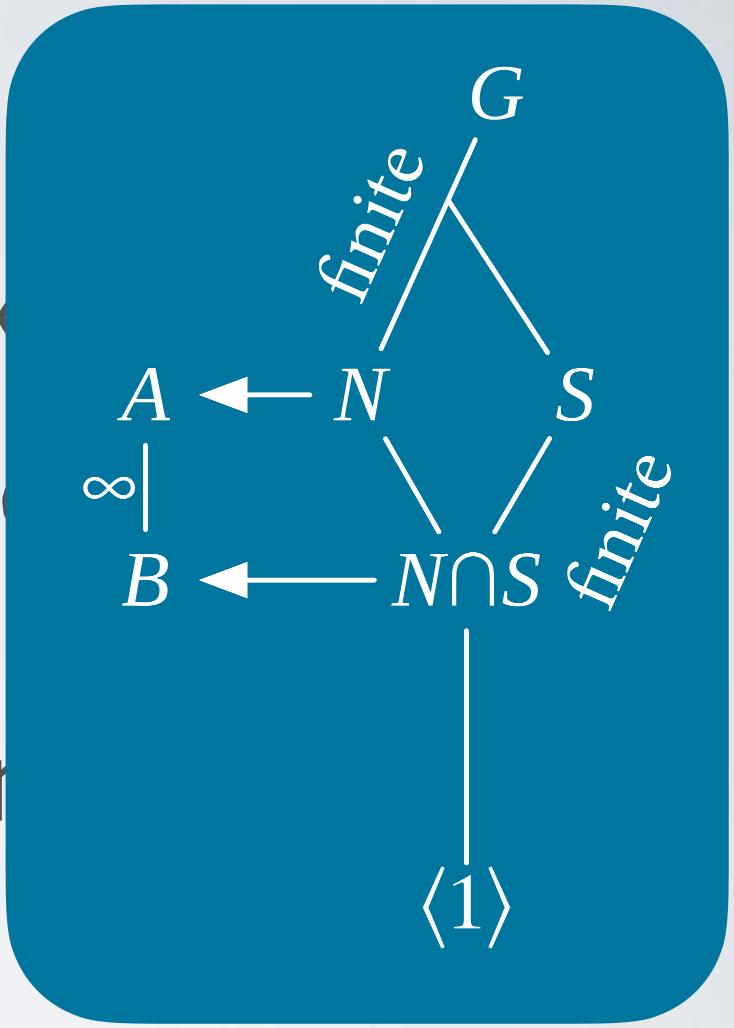
Some attempts of proving infinite index:

- Find an automatic structure that proves infinite index. (Requires suitable embedding.)
  Normal forms for subgroup elements might show that there are infinitely many cosets.
- Show that one can find an infinite index in a suitable quotient image of normal subgroup. (Applicable, if CSP does not hold.)

Proving Infinite Index

Some attempts of proving infinite index

- Find an automatic structure that prov index. (Requires suitable embedding.) Normal forms for subgroup elements r that there are infinitely many cosets.



- Show that one can find an infinite index in a suitable quotient image of normal subgroup. (Applicable, if CSP does not hold.)

$$\left\langle \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} \right\rangle \leq \mathrm{SL}_2(\mathbb{Z})$$

```
gap> psi:=SLNZFP(2);;F:=Source(psi);
gap> g:=Group([[1,3],[0,1]],[[1,0],[3,1]]);
gap> w:=List(GeneratorsOfGroup(g),x->PreImagesRepresentative(psi,x));
[ T^3, S*T^-2*S*T*S^-1*T ]
gap> LoadPackage("kbmag");
Loading  kbmag 1.5.9 by Derek Holt
gap> R:=KBMAGRewritingSystem(F);;
gap> S:=SubgroupOfKBMAGRewritingSystem(R,w);
gap> A:=AutomaticStructureOnCosetsWithSubgroupPresentation(R, S);
true
gap> Index(R,S);
infinity # proof!
```

$$\left\langle \begin{pmatrix}1 & 3\\ 0 & 1\end{pmatrix}, \begin{pmatrix}1 & 0\\ 3 & 1\end{pmatrix} \right\rangle \leq \mathrm{SL}_2(\mathbb{Z})$$

```
gap> psi:=SLNZFP(2);;F:=Source(psi);
gap> g:=Group([[1,3],[0,1]],[[1,0],[3,1]]);
gap> w:=List(GeneratorsOfGroup(g),x->PreImagesRepresentative(psi,x));
[ T^3, S*T^-2*S*T*S^-1*T ]
gap> LoadPackage("kbmag");
Loading  kbmag 1.5.9 by Derek Holt
gap> R:=KBMAGRewritingSystem(F);;
gap> S:=SubgroupOfKBMAGRewritingSystem(R,w);
gap> A:=AutomaticStructureOnCosetsWithSubgroupPresentation(R, S);
true
gap> Index(R,S);
infinity # proof!
```

Might `fail`/`false`

Q: Can this work in dimension >2 ?

Example Second Strategy

$R = \mathbb{Z}(\zeta), \zeta^2 + \zeta = -1,$

$\Gamma = GL_2(R) = \langle t, u, j, l, a, w \rangle$ with

$$t = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, u = \begin{pmatrix} 1 & \zeta \\ 0 & 1 \end{pmatrix}, j = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, l = \begin{pmatrix} \zeta^2 & 0 \\ 0 & \zeta \end{pmatrix}, a = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, w = \begin{pmatrix} -\zeta & 0 \\ 0 & 1 \end{pmatrix}$$

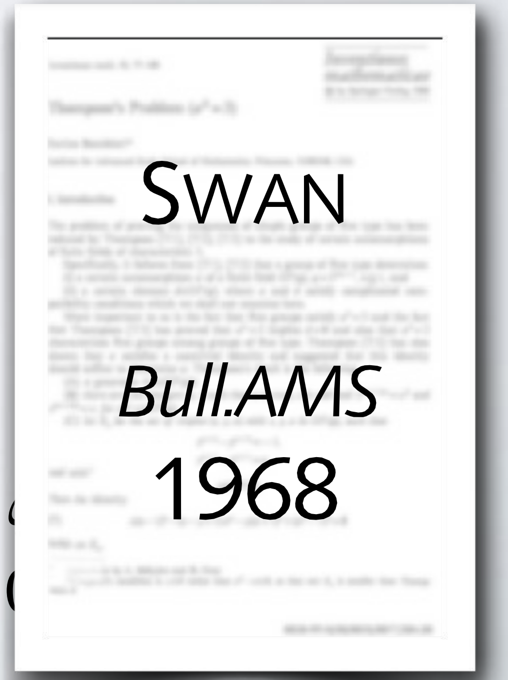
with relations $tu = ut, j^2, tj = jt, uj = ju, lj = jl$, $aj = ja$,$l^3, l^{-1}tl = t^{-1}u^{-1}, l^{-1}ul = t, a^2 = j,$ $(al)^2 = j, (ta)^3 = j, (ual)^3 = j, wj = jw, w^6,$ $wtw^{-1} = u^{-1}, wuw^{-1} = tu, waw^{-1} = jl^2a, wl = lw$

MARGOLIS/BÄCHLE: Is $[\Gamma : G]$ finite for

$G = \langle m_1, m_2, m_3, m_i, m_j, m_t \rangle$ with

$$m_1 = \begin{pmatrix} 97\zeta^2 & -112\zeta - 56\zeta^2 \\ 112\zeta + 56\zeta^2 & 97\zeta^2 \end{pmatrix}, m_2 = \begin{pmatrix} 56\zeta + 41\zeta^2 & 56\zeta + 112\zeta^2 \\ 56\zeta + 112\zeta^2 & -56\zeta + 153\zeta^2 \end{pmatrix},$$

$$m_3 = \begin{pmatrix} 56\zeta + 209\zeta^2 & -56\zeta + 56\zeta^2 \\ -56\zeta + 56\zeta^2 & -56\zeta - 15\zeta^2 \end{pmatrix}, m_i = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, m_j = \begin{pmatrix} \zeta & \zeta^2 \\ \zeta^2 & -\zeta \end{pmatrix}, m_t =$$



SWAN

Bull.AMS

1968

Example Second Strategy

$R = \mathbb{Z}(\zeta), \zeta^2 + \zeta = -1,$

$\Gamma = GL_2(R) = \langle t, u, j, l, a, w \rangle$ with

$t = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, u = \begin{pmatrix} 1 & \zeta \\ 0 & 1 \end{pmatrix}, j = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, l = \begin{pmatrix} \zeta^2 & 0 \\ 0 & \zeta \end{pmatrix}, a = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, w = \begin{pmatrix} -\zeta & 0 \\ 0 & 1 \end{pmatrix}$

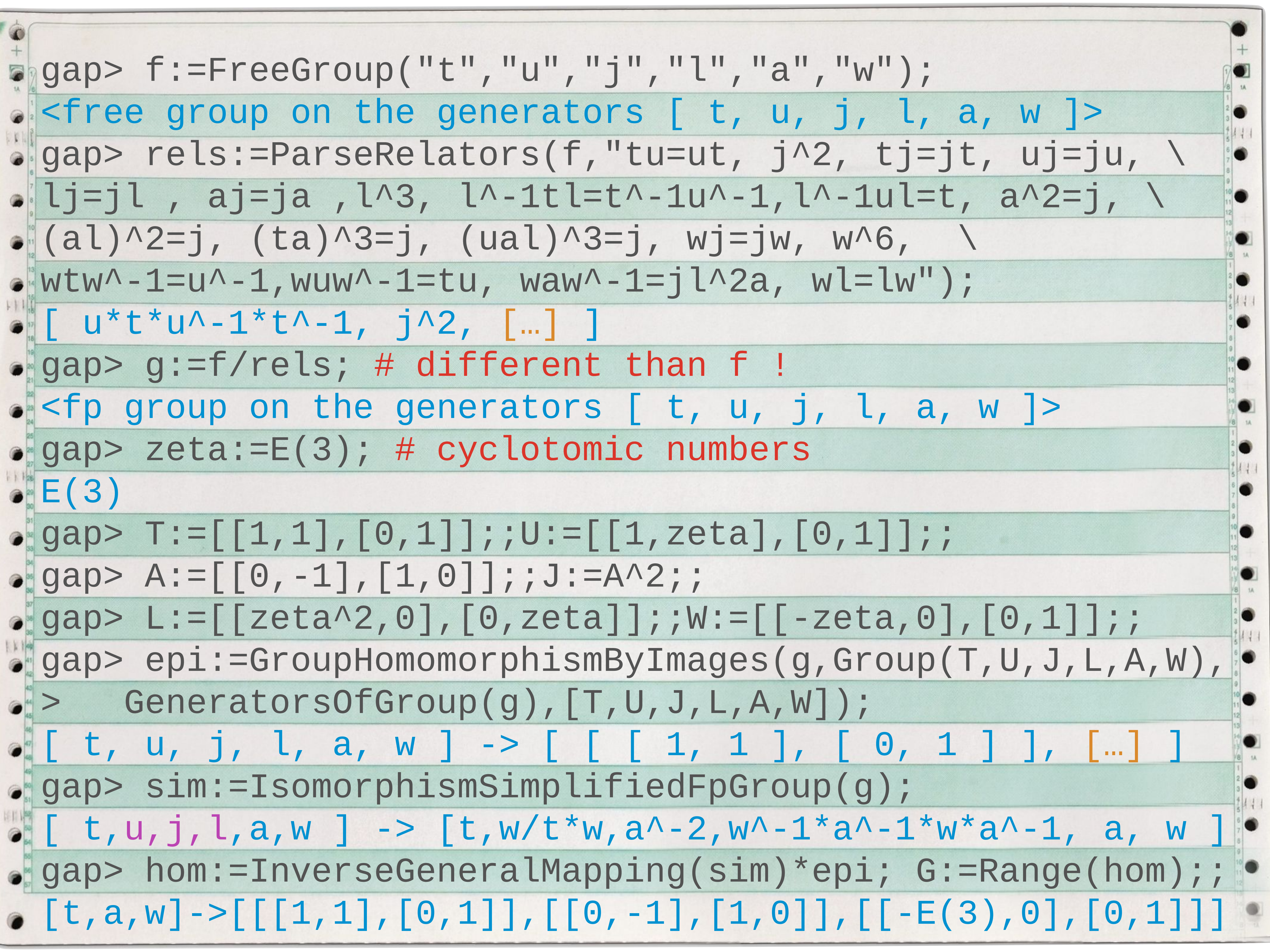
with relations $tu = ut, j^2, tj = jt, uj = ju, lj = jl$,
$aj = ja$,$l^3, l^{-1}tl = t^{-1}u^{-1}, l^{-1}ul = t, a^2 = j,$
$(al)^2 = j, (ta)^3 = j, (ual)^3 = j, wj = jw, w^6,$
$wtw^{-1} = u^{-1}, wuw^{-1} = tu, waw^{-1} = jl^2a, wl = lw$

MARGOLIS/BÄCHLE: Is $[\Gamma : G]$ finite for

$G = \langle m_1, m_2, m_3, m_i, m_j, m_t \rangle$ with

$m_1 = \begin{pmatrix} 97\zeta^2 & -112\zeta - 56\zeta^2 \\ 112\zeta + 56\zeta^2 & 97\zeta^2 \end{pmatrix}, m_2 = \begin{pmatrix} 56\zeta + 41\zeta^2 & 56\zeta + 112\zeta^2 \\ 56\zeta + 112\zeta^2 & -56\zeta + 153\zeta^2 \end{pmatrix},$

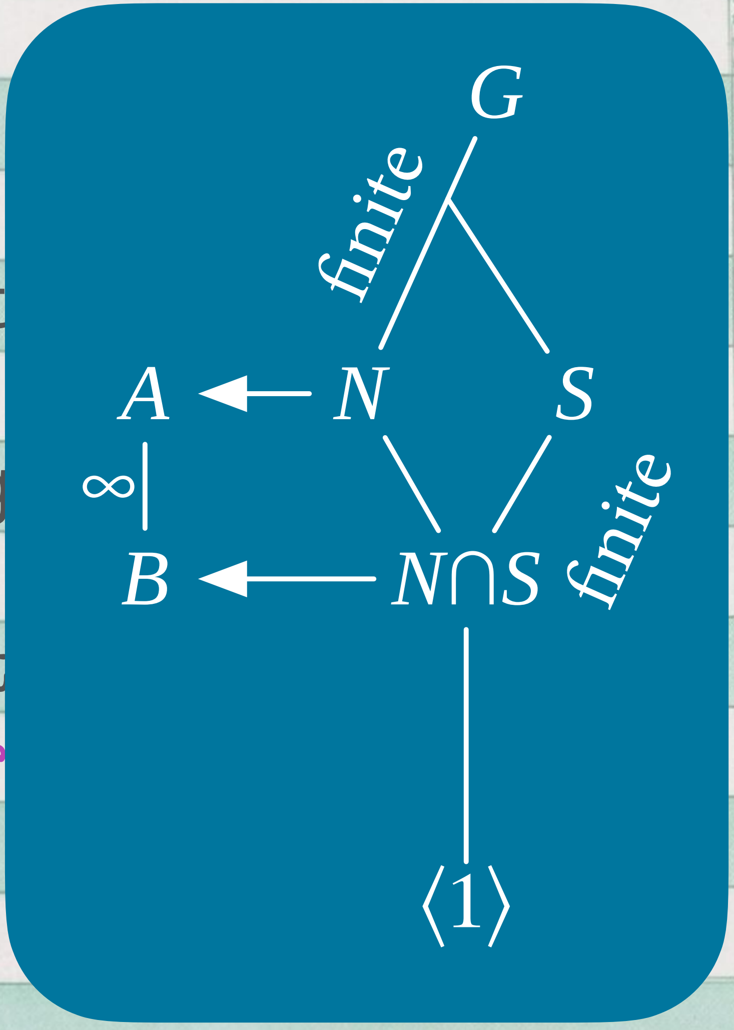
$m_3 = \begin{pmatrix} 56\zeta + 209\zeta^2 & -56\zeta + 56\zeta^2 \\ -56\zeta + 56\zeta^2 & -56\zeta - 15\zeta^2 \end{pmatrix}, m_i = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, m_j = \begin{pmatrix} \zeta & \zeta^2 \\ \zeta^2 & -\zeta \end{pmatrix}, m_t = \begin{pmatrix} \zeta & 0 \\ 0 & \zeta \end{pmatrix}$

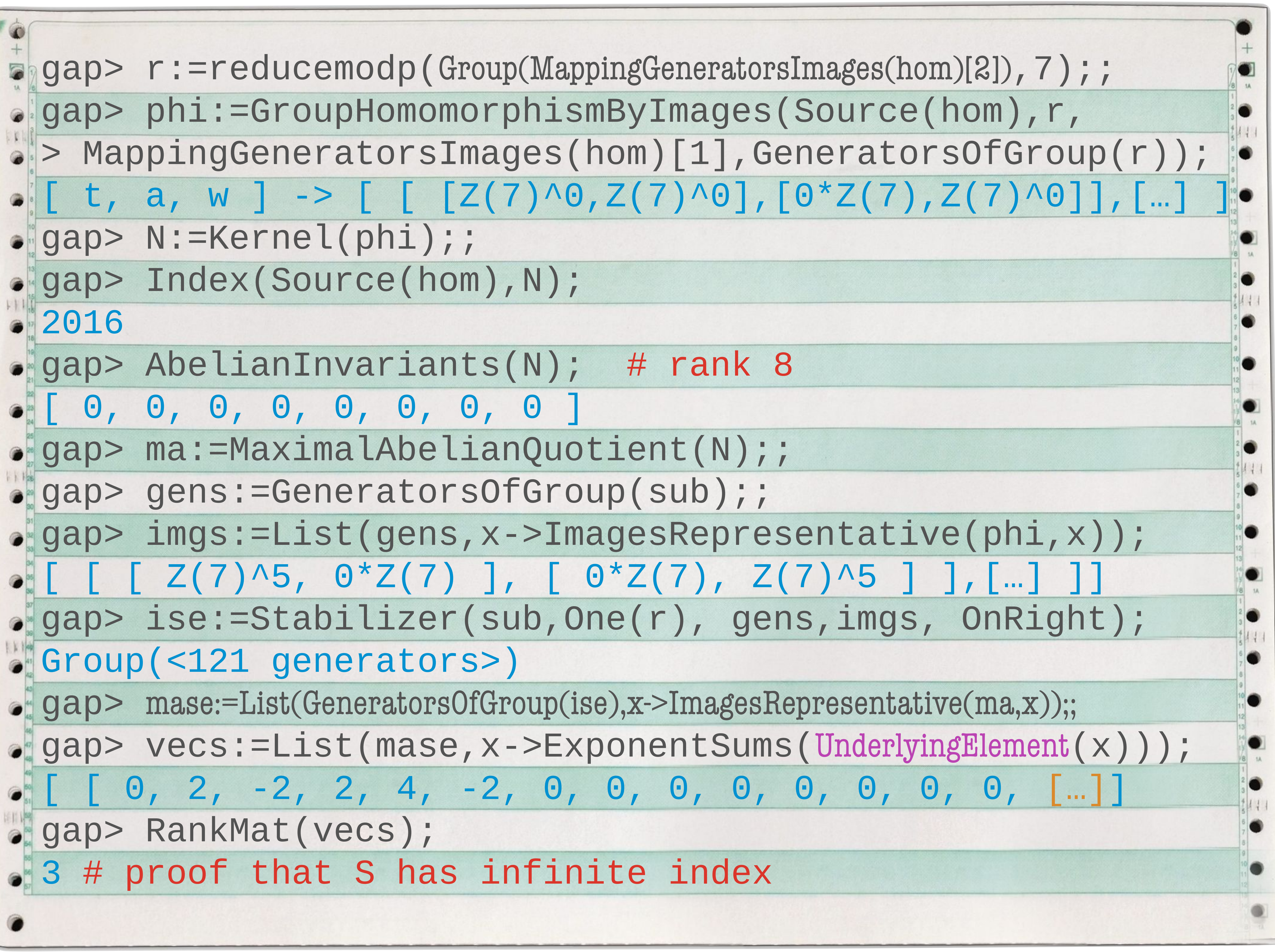


```
gap> f:=FreeGroup("t","u","j","l","a","w");
<free group on the generators [ t, u, j, l, a, w ]>
gap> rels:=ParseRelators(f,"tu=ut, j^2, tj=jt, uj=ju, \
lj=jl , aj=ja ,l^3, l^-1tl=t^-1u^-1,l^-1ul=t, a^2=j, \
(al)^2=j, (ta)^3=j, (ual)^3=j, wj=jw, w^6,   \
wtw^-1=u^-1,wuw^-1=tu, waw^-1=jl^2a, wl=lw");
[ u*t*u^-1*t^-1, j^2, […] ]
gap> g:=f/rels; # different than f !
<fp group on the generators [ t, u, j, l, a, w ]>
gap> zeta:=E(3); # cyclotomic numbers
E(3)
gap> T:=[[1,1],[0,1]];;U:=[[1,zeta],[0,1]];;
gap> A:=[[0,-1],[1,0]];;J:=A^2;;
gap> L:=[[zeta^2,0],[0,zeta]];;W:=[[-zeta,0],[0,1]];;
gap> epi:=GroupHomomorphismByImages(g,Group(T,U,J,L,A,W),
>    GeneratorsOfGroup(g),[T,U,J,L,A,W]);
[ t, u, j, l, a, w ] -> [ [ [ 1, 1 ], [ 0, 1 ] ], […] ]
gap> sim:=IsomorphismSimplifiedFpGroup(g);
[ t,u,j,l,a,w ] -> [t,w/t*w,a^-2,w^-1*a^-1*w*a^-1, a, w ]
gap> hom:=InverseGeneralMapping(sim)*epi; G:=Range(hom);;
[t,a,w]->[[[1,1],[0,1]],[[0,-1],[1,0]],[[-E(3),0],[0,1]]]
```

```
gap> normred:=function(phi,mat) […] end;;
gap> w1:=normred(hom,m1);
((t^-1*w^-1*a^-1*t^-1)^8)^(w^-1)
gap> w2:=normred(hom,m2);; w3:=normred(hom,m3);;
gap> wi:=normred(hom,mi);; wj:=normred(hom,mj);;
gap> wt:=normred(hom,mt);;
gap> rg:=Source(hom);;
gap> sub:=Subgroup(rg,[w1,w2,w3,wi,wj,wt]);;# fp version
gap> reducemodp:=function(g,p) […] end;;
gap> prd:=3*7*31*97*169*361;  # found by trying out
3852535323
gap> h:=reducemodp(G,prd);
Group([[[ZmodnZObj( 1, 3852535323 ), […]  )
gap> FittingFreeLiftSetup(h);;Size(h);
295803518081960107436094259200
gap> hs:=reducemodp(S,prd);;FittingFreeLiftSetup(hs);;
gap> Size(h)/Size(hs);
142580874240
```

```
gap> r:=reducemodp(Group(MappingGeneratorsImages(hom)[2]),7);;
gap> phi:=GroupHomomorphismByImages(Source(hom),r,
> MappingGeneratorsImages(hom)[1],GeneratorsOfGroup(r));
[ t, a, w ] -> [ [ [Z(7)^0,Z(7)^0],[0*Z(7),Z(7)^0]],[…] ]
gap> N:=Kernel(phi);;
gap> Index(Source(hom),N);
2016
gap> AbelianInvariants(N);  # rank 8
[ 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> ma:=MaximalAbelianQuotient(N);;
gap> gens:=GeneratorsOfGroup(sub);;
gap> imgs:=List(gens,x->ImagesRepresentat
[ [ [ Z(7)^5, 0*Z(7) ], [ 0*Z(7), Z(7)^5
gap> ise:=Stabilizer(sub,One(r), gens,img
Group(<121 generators>)
gap> mase:=List(GeneratorsOfGroup(ise),x->ImagesRepresent
gap> vecs:=List(mase,x->ExponentSums(Under
[ [ 0, 2, -2, 2, 4, -2, 0, 0, 0, 0, 0, 0,
gap> RankMat(vecs);
3 # proof that S has infinite index
```





```
gap> r:=reducemodp(Group(MappingGeneratorsImages(hom)[2]),7);;
gap> phi:=GroupHomomorphismByImages(Source(hom),r,
> MappingGeneratorsImages(hom)[1],GeneratorsOfGroup(r));
[ t, a, w ] -> [ [ [Z(7)^0,Z(7)^0],[0*Z(7),Z(7)^0]],[…] ]
gap> N:=Kernel(phi);;
gap> Index(Source(hom),N);
2016
gap> AbelianInvariants(N);  # rank 8
[ 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> ma:=MaximalAbelianQuotient(N);;
gap> gens:=GeneratorsOfGroup(sub);;
gap> imgs:=List(gens,x->ImagesRepresentative(phi,x));
[ [ [ Z(7)^5, 0*Z(7) ], [ 0*Z(7), Z(7)^5 ] ],[…] ]]
gap> ise:=Stabilizer(sub,One(r), gens,imgs, OnRight);
Group(<121 generators>)
gap> mase:=List(GeneratorsOfGroup(ise),x->ImagesRepresentative(ma,x));;
gap> vecs:=List(mase,x->ExponentSums(UnderlyingElement(x)));
[ [ 0, 2, -2, 2, 4, -2, 0, 0, 0, 0, 0, 0, 0, 0, […]]
gap> RankMat(vecs);
3 # proof that S has infinite index
```