

Mini-course on GAP – Lecture 3

Leandro Vendramin

Vrije Universiteit Brussel

March 2023



SmallGroups

GAP contains a [database with all groups of certain small orders](#). The groups are sorted by their orders and they are listed up to isomorphism. This database is part of a library named `SmallGroups`. It contains the following groups:

- ▶ those of order ≤ 2000 except order 1024,
- ▶ those of cube-free order ≤ 50000 ,
- ▶ those of order p^7 for $p \in \{3, 5, 7, 11\}$,
- ▶ those of order p^n for $n \leq 6$ and all primes p ,
- ▶ those of order $q^n p$ for q^n dividing 2^8 , 3^6 , 5^5 or 7^4 and all primes p with $p \neq q$,
- ▶ those of square-free order.

The library was written by H. Besche, B. Eick and E. O'Brien.

Do you want to see what GAP knows about groups of order twelve?
Just use the function `SmallGroupsInformation`.

There exist non-abelian groups of odd order and that the smallest of this group has order 21:

```
gap> First(AllSmallGroups(Size, [1, 3..21]),\
> x->not IsAbelian(x));;
gap> Size(last);
21
```

There are no simple groups of order 84. We use the filter `IsSimple` with the function `AllSmallGroups`:

```
gap> AllSmallGroups(Size, 84, IsSimple, true);  
[ ]
```

With the function `StructureDescription` one explores the [structure of a given group](#). The function returns a short string which gives some insight into the structure of the group. Let us see how the groups of order twelve look like:

```
gap> List(AllSmallGroups(Size, 12),\
> StructureDescription);
[ "C3 : C4", "C12", "A4", "D12", "C6 x C2" ]
```

The group `C3 : C4` denotes the semidirect product $C_3 \rtimes C_4$.

The string returned by `StructureDescription` is not an isomorphism invariant: non-isomorphic groups can have the same string value and two isomorphic groups in different representations can produce different strings.

There are two groups of order 20 that can be written as a semidirect product $C_5 \rtimes C_4$. `StructureDescription` will not distinguish such groups:

```
gap> List(AllSmallGroups(Size, 20),\
> StructureDescription);
[ "C5 : C4", "C20", "C5 : C4", "D20", "C10 x C2" ]
```

SmallGroups

To [identify groups](#) in the database SmallGroups one uses the function `IdGroup`.

```
gap> IdGroup(SymmetricGroup(3));  
[ 6, 1 ]  
gap> IdGroup(SymmetricGroup(4));  
[ 24, 12 ]  
gap> IdGroup(AlternatingGroup(4));  
[ 12, 3 ]  
gap> IdGroup(DihedralGroup(8));  
[ 8, 3 ]  
gap> IdGroup(QuaternionGroup(8));  
[ 8, 4 ]
```

Lam and Leep¹ proved that each index-two subgroup of $\text{Aut}(\text{Sym}_6)$ is isomorphic either to Sym_6 , $\mathbf{PGL}_2(9)$ or to the Mathieu group M_{10} . Let us check this claim using the function `IdGroup`:

```
gap> autS6 := AutomorphismGroup(SymmetricGroup(6));;
gap> lst := SubgroupsOfIndexTwo(autS6);;
gap> List(lst, IdGroup);
[ [ 720, 764 ], [ 720, 763 ], [ 720, 765 ] ]
gap> IdGroup(PGL(2,9));
[ 720, 764 ]
gap> IdGroup(MathieuGroup(10));
[ 720, 765 ]
gap> IdGroup(SymmetricGroup(6));
[ 720, 763 ]
```

¹Exposition. Math. 11 (1993), no. 4, 289–308

Guralnick's theorem on commutators

Guralnick² proved without using computers that the **smallest group** G such that $[G, G] \neq \{[x, y] : x, y \in G\}$ has order 96. Here is the proof:

```
gap> G := First(AllSmallGroups(Size, [1..100]), \
> x->Order(DerivedSubgroup(x))<>Size(\
> Set(List(Cartesian(x,x), Comm))));;
gap> Order(G);
96
```

²Adv. in Math., 45(3):319–330, 1982

Guralnick's theorem on commutators

With `IdGroup` (or with `IsomorphismGroups`) we can check that

$$G \simeq \langle (135)(246)(7\ 11\ 9)(8\ 12\ 10), (394\ 10)(58)(67)(11\ 12) \rangle.$$

```
gap> IdGroup(G);  
[ 96, 3 ]  
gap> a := (1,3,5)(2,4,6)(7,11,9)(8,12,10);;  
gap> b := (3,9,4,10)(5,8)(6,7)(11,12);;  
gap> IdGroup(Group([a,b]));  
[ 96, 3 ]
```

Okay, but how did we find this isomorphism?

Guralnick's theorem on commutators

We have our group G . We use the function `IsomorphismPermGroup` to construct a faithful representation of G as a permutation group. With `SmallerDegreePermutationRepresentation` we construct (if possible) an isomorphic permutation group of smaller degree. Be aware that this new degree may not be minimal. After some attempts, we obtain an isomorphic copy of G inside Sym_{12} . To construct a set of generators we then use `SmallGeneratingSet`. Again, be aware that this set may not be minimal.

Can you try this yourself? Be aware that maybe you will not get the exact same result.

A theorem of Navarro

For a finite group G let $cs(G)$ denote the set of sizes of the conjugacy classes of G , that is

$$cs(G) = \{|g^G| : g \in G\}.$$

For example: $cs(\text{Sym}_3) = \{1, 2, 3\}$ and $cs(\mathbf{SL}_2(3)) = \{1, 4, 6\}$.

```
gap> cs := function(group)
> return Set(List(ConjugacyClasses(group), Size));
> end;
function( group ) ... end
gap> cs(SymmetricGroup(3));
[ 1, 2, 3 ]
gap> cs(SL(2,3));
[ 1, 4, 6 ]
```

A theorem of Navarro

We will write $G_{n,k}$ to denote the k -th group of size n in the database, thus $G_{n,k}$ is a group with `IdGroup` equal to $[n, k]$.

A theorem of Navarro

Navarro³ proved that there exist finite groups G and H such that G is solvable, H is not solvable and $\text{cs}(G) = \text{cs}(H)$. This answers a question of Brauer.

Let $G = G_{240,13} \times G_{960,1019}$ and $H = G_{960,239} \times G_{480,959}$. Then G is solvable, H is not solvable and $\text{cs}(G) = \text{cs}(H)$.

```
gap> U := SmallGroup(960,239);;  
gap> V := SmallGroup(480,959);;  
gap> L := SmallGroup(960,1019);;  
gap> K := SmallGroup(240,13);;  
gap> UxV := DirectProduct(U,V);;  
gap> KxL := DirectProduct(K,L);;  
gap> IsSolvable(UxV);  
false  
gap> IsSolvable(KxL);  
true
```

³J. Algebra 411 (2014), 47–49.

A theorem of Navarro

One could try to compute $cs(U \times V)$ directly. However, this calculation seems to be hard. The trick is to use that

$$cs(U \times V) = \{nm : n \in cs(U), m \in cs(V)\}.$$

```
gap> cs(KxL)=Set(List(Cartesian(cs(U),cs(V)),\
> x->x[1]*x[2]));
true
```

Another theorem of Navarro

Navarro proved that there exist finite groups G and H such that G is nilpotent, $Z(H) = 1$ and $\text{cs}(G) = \text{cs}(H)$. This answers another question of Brauer.

The groups are $G = \mathbb{D}_8 \times G_{243,26}$ and $H = G_{486,36}$.

```
gap> K := DihedralGroup(8);;
gap> L := SmallGroup(243,26);;
gap> H := SmallGroup(486,36);;
gap> IsTrivial(Center(H));
true
gap> G := DirectProduct(K,L);;
gap> cs(G)=cs(H);
true
gap> IsNilpotent(G);
true
```

Finitely presented groups

Let us start working with **free groups**. The function `FreeGroup` constructs the free group in a finite number of generators. We create the free group F_2 in two generators and we create some **random elements** with the function `Random`:

```
gap> f := FreeGroup(2);  
<free group on the generators [ f1, f2 ]>  
gap> f.1^2;  
f1^2  
gap> f.1^2*f.1;  
f1^3  
gap> f.1*f.1^(-1);  
<identity ...>  
gap> Random(f);  
f1^-3
```

Finitely presented groups

The function `Length` can be used to compute the **length of words** in a free group. In this example we create 10000 random elements in F_2 and compute their lengths.

```
gap> f := FreeGroup(2);;
gap> Collected(List(List([1..10000],\
> x->Random(f)), Length));
[ [ 0, 2270 ], [ 1, 1044 ], [ 2, 1113 ],
  [ 3, 986 ], [ 4, 874 ], [ 5, 737 ],
  [ 6, 642 ], [ 7, 500 ], [ 8, 432 ],
  [ 9, 329 ], [ 10, 248 ], [ 11, 189 ],
  [ 12, 152 ], [ 13, 119 ], [ 14, 93 ],
  [ 15, 68 ], [ 16, 57 ], [ 17, 34 ],
  [ 18, 30 ], [ 19, 23 ], [ 20, 19 ],
  [ 21, 16 ], [ 22, 8 ], [ 23, 3 ], [ 24, 4 ],
  [ 25, 4 ], [ 26, 2 ], [ 27, 2 ], [ 28, 1 ],
  [ 31, 1 ] ]
```

Finitely presented groups

Some of the functions we used before can also be used in free groups. Examples of these functions are `Normalizer`, `RepresentativeAction`, `IsConjugate`, `Intersection`, `IsSubgroup`, `Subgroup`.

The free group F_2

Here we perform some elementary calculations in F_2 , the free group with generators a and b .

```
gap> f := FreeGroup("a", "b");;
gap> a := f.1;;
gap> b := f.2;;
gap> Random(f);
b^-1*a^-5
gap> Centralizer(f, a);
Group([ a ])
gap> Index(f, Centralizer(f, a));
infinity
gap> Subgroup(f, [a,b]);
Group([ a, b ])
gap> Order(Subgroup(f, [a,b]));
infinity
```

The free group F_2

We compute the automorphism group of F_2 .

```
gap> AutomorphismGroup(f);  
<group of size infinity with 3 generators>  
gap> GeneratorsOfGroup(AutomorphismGroup(f));  
[ [ a, b ] -> [ a^-1, b ],  
  [ a, b ] -> [ b, a ],  
  [ a, b ] -> [ a*b, b ] ]
```

The free group F_2

We now check that the subgroup S generated by a^2 , b and aba^{-1} has index two in F_2 . We compute $\text{Aut}(S)$ and check that it is not a free group:

```
gap> S := Subgroup(f, [a^2, b, a*b*a^(-1)]);  
Group([ a^2, b, a*b*a^-1 ])  
gap> Index(f, S);  
2  
gap> A := AutomorphismGroup(S);  
<group of size infinity with 3 generators>  
gap> IsFreeGroup(A);  
false
```

Finitely presented groups

The group

$$G = \langle a, b, c : ba = ac, ca = ab, bc = ca \rangle$$

has an infinite number of elements and its center has finite index.

```
gap> f := FreeGroup(3);;
gap> a := f.1;;
gap> b := f.2;;
gap> c := f.3;;
gap> gr := f/[a^b*Inverse(c),\
> a^c*Inverse(b),\
> b^c*Inverse(a)];;
gap> Order(gr);
infinity
gap> Center(gr);
Group([ f2^2 ])
gap> StructureDescription(gr/Center(gr));
"S3"
```

Finitely presented groups

The abelianization of G is isomorphic to \mathbb{Z} .

```
gap> gr/DerivedSubgroup(gr);  
Group([ f1*f2^-1*f3, f3, f2^-1*f3 ])  
gap> AbelianInvariants(gr/DerivedSubgroup(gr));  
[ 0 ]
```

Since the index $(G : Z(G))$ is finite, a theorem of Schur implies that the commutator subgroup $[G, G]$ is a finite group. However, GAP cannot prove this!

A theorem of Coxeter

Let $n \geq 3$ and $p \geq 2$ be integers. Coxeter⁴ proved that the group generated by $\sigma_1, \dots, \sigma_{n-1}$ and

$$\begin{array}{ll} \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} & \text{if } i \in \{1, \dots, n-2\}, \\ \sigma_i \sigma_j = \sigma_j \sigma_i & \text{if } |i-j| \geq 2, \\ \sigma_i^p = 1 & \text{if } i \in \{1, \dots, n-1\}, \end{array}$$

is finite if and only if $(p-2)(n-2) < 4$.

⁴Kaleidoscopes. Selected writings of H. S. M. Coxeter.

A theorem of Coxeter

We study the case $n = 3$. Let

$$G = \langle a, b : aba = bab, a^p = b^p = 1 \rangle.$$

We claim that

$$G \simeq \begin{cases} \text{Sym}_3 & \text{if } p = 2, \\ \mathbf{SL}_2(3) & \text{if } p = 3, \\ \mathbf{SL}_2(3) \rtimes C_4 & \text{if } p = 4, \\ \mathbf{SL}_2(3) \rtimes C_5 & \text{if } p = 5 : \end{cases}$$

A theorem of Coxeter

Here is the proof:

```
gap> f := FreeGroup(2);;
gap> a := f.1;;
gap> b := f.2;;
gap> p := 2;;
gap> while p-2<4 do
> G := f/[a*b*a*Inverse(b*a*b), a^p, b^p];;
> Display(StructureDescription(G));
> p := p+1;
> od;
S3
SL(2,3)
SL(2,3) : C4
C5 x SL(2,5)
```

A theorem of von Dyck

For $l, m, n \in \mathbb{N}$, we define the **von Dyck group** (or triangular group) of type (l, m, n) as the group

$$G(l, m, n) = \langle a, b : a^l = b^m = (ab)^n = 1 \rangle.$$

It is known that $G(l, m, n)$ is finite if and only if

$$\frac{1}{l} + \frac{1}{m} + \frac{1}{n} > 1.$$

We claim that

$$G(2, 3, 3) \simeq \text{Alt}_4, \quad G(2, 3, 4) \simeq \text{Sym}_4, \quad G(2, 3, 5) \simeq \text{Alt}_5.$$

A theorem of von Dyck

Here is the proof:

```
gap> f := FreeGroup(2);;
gap> a := f.1;;
gap> b := f.2;;
gap> StructureDescription(f/[a^2,b^3,(a*b)^3]);
"A4"
gap> StructureDescription(f/[a^2,b^3,(a*b)^4]);
"S4"
gap> StructureDescription(f/[a^2,b^3,(a*b)^5]);
"A5"
```

Some presentations of the trivial group

This example is taken from Pierre de la Harpe's book⁵. The group

$$\langle a, b, c : a^3 = b^3 = c^4 = 1, ac = ca^{-1}, aba^{-1} = bcb^{-1} \rangle$$

is trivial.

```
gap> f := FreeGroup(3);;
gap> a := f.1;;
gap> b := f.2;;
gap> c := f.3;;
gap> G := f/[a^3, b^3, c^4, c^(-1)*a*c*a, \
> a*b*a^(-1)*b*c^(-1)*b^(-1)];;
gap> IsTrivial(G);
true
```

⁵Topics in geometric group theory.

Some presentations of the trivial group

Miller and Schupp⁶ proved that for $n \in \mathbb{N}$,

$$\langle a, b : a^{-1}b^na = b^{n+1}, a = a^{i_1}b^{j_1}a^{i_2}b^{j_2} \dots a^{i_k}b^{j_k} \rangle,$$

is trivial if $i_1 + i_2 + \dots + i_k = 0$. As an example let us see that

$$\langle a, b : a^{-1}b^2a = b^3, a = a^{-1}ba \rangle$$

is the trivial group:

```
gap> f := FreeGroup(2);;
gap> a := f.1;;
gap> b := f.2;;
gap> G := f/[a^(-1)*b^2*a*b^(-3), a*(a^(-1)*b*a)];;
gap> IsTrivial(G);
true
```

⁶Groups, languages and geometry, 113–115, Contemp. Math., 250, 1999.

Burnside problem

For each $n \geq 2$ the Burnside group $B(2, n)$ is defined as the group

$$B(2, n) = \langle a, b : w^n = 1 \text{ for all word } w \text{ in the letters } a \text{ and } b \rangle.$$

Is the group $B(2, n)$ finite?

The particular case $B(2, 5)$ remains open.

Burnside problem: A theorem of Burnside

We prove that the group $B(2, 3)$ is a finite group of order ≤ 27 . Let F be the free group of rank two. We divide F by the normal subgroup generated by $\{w_1^3, \dots, w_{10000}^3\}$, where w_1, \dots, w_{10000} are some randomly chosen words of F . The following code shows that $B(2, 3)$ is finite:

```
gap> f := FreeGroup(2);;
gap> rels := Set(List([1..10000], \
> x->Random(f)^3));;
gap> G := f/rels;;
gap> Order(G);
27
```

Burnside problem: A theorem of Sanov

It is known that $B(2, 4)$ is a finite group. Here we present here a computational proof. We use the same trick as before to prove that $B(2, 4)$ is finite and has order ≤ 4096 :

```
gap> f := FreeGroup(2);;
gap> rels := Set(List([1..10000], \
> x->Random(f)^4));;
gap> B24 := f/rels;;
gap> Order(B24);
4096
```

A problem by Djokovic

In 1970 Djokovic posed in the *Canadian Mathematical Bulletin* the following problem: Prove that the alternating groups Alt_n for $n \geq 5$ and $n \neq 8$ can be generated by three conjugate involutions. In his solution, published in the *Canadian Mathematical Bulletin* in 1972, he writes that he does not know what happens if $n = 8$.

A problem by Djokovic

We write a function that finds all possible conjugate involutions that generate the whole group. The code written will be is pretty naive, one just runs (in a clever way) over all subsets of three conjugate involutions and checks whether these three permutation generate the whole group.

A problem by Djokovic

```
gap> Djokovic := function(n)
> local gr, cc, c, t, l;
> l := [];
> gr := AlternatingGroup(n);;
> cc := ConjugacyClasses(gr);;
> for c in cc do
>   if Order(Representative(c))=2 then
>     for t in IteratorOfCombinations(AsList(c), 3) do
>       if Size(Subgroup(gr, t))=Size(gr) then
>         Add(l, t);
>       fi;
>     od;
>   fi;
> od;
> return l;
> end;
function( n ) ... end
```

A problem by Djokovic

We see that Alt_5 can be generated by the involutions $(23)(45)$, $(24)(35)$ and $(12)(45)$:

```
gap> Djokovic(5)[1];  
[ (2,3)(4,5), (2,4)(3,5), (1,2)(4,5) ]
```

There are 380 generating sets that fit into Djokovic assumptions:

```
gap> Size(Djokovic(5));  
380
```

A problem by Djokovic

Finally we prove we cannot find three conjugate involutions of Alt_8 that generate the whole Alt_8 . The calculation is straightforward but requires several minutes to be performed:

```
gap> Djokovic(8);  
[ ]
```

A theorem of Dixon

The **commuting probability** of a finite group G is defined as the probability that a randomly chosen pair of elements of G commute, and it is thus equal to $k(G)/|G|$. The following function computes the commuting probability of a given finite group.

```
gap> p := x->NrConjugacyClasses(x)/Order(x);  
function( x ) ... end
```

Dixon observed that the commuting probability of a finite non-abelian simple group is $\leq 1/12$. This bound is attained for the alternating simple group Alt_5 .

```
gap> p(AlternatingGroup(5));  
1/12
```

A theorem of Dixon

One can find Dixon's proof in a 1973 volume of the *Canadian Mathematical Bulletin*. The proof we present here was found by Iván Sadofski Costa.

We first assume that the commuting probability of G is $> 1/12$. Since G is a non-abelian simple group, the identity is the only central element. Let us assume first that there is a conjugacy class of G of size m , where m is such that $1 < m \leq 12$. Then G is a transitive subgroup of Sym_m .

A **transitive group** of degree n is a subgroup of Sym_n that acts transitively on $\{1, \dots, n\}$; in this case, n is the degree of the transitive group. GAP contains a database with all transitive groups of low degree.

Now the problem is easy: we show that there are no non-abelian simple groups that act transitively on sets of size $m \in \{2, \dots, 12\}$ with commuting probability $> 1/12$.

A theorem of Dixon

```
gap> l := AllTransitiveGroups(NrMovedPoints, \
> [2..12], \
> IsAbelian, false, \
> IsSimple, true);;
gap> List(l, p);
[ 1/12, 1/12, 7/360, 1/28, 1/280, 1/28, 1/1440,
  1/56, 1/10080, 1/12, 7/360, 1/75600, 2/165,
  1/792, 31/19958400, 1/12, 2/165, 1/792, 1/6336,
  43/239500800 ]
gap> ForAny(l, x->p(x)>1/12);
false
```

A theorem of Dixon

Now assume that all non-trivial conjugacy class of G have at least 13 elements. Then the class equation implies that

$$|G| \geq \frac{13}{12}|G| - 12,$$

and therefore $|G| \leq 144$. Thus one needs to check what happens with groups of order ≤ 144 . But we know that the only non-abelian simple group of size ≤ 144 is the alternating simple group Alt_5 .

```
gap> AllGroups(Size, [2..144], \
> IsAbelian, false, \
> IsSimple, true);
[ Alt( [ 1 .. 5 ] ) ]
```

An exercise on primitive groups

A subgroup G of Sym_n is called **primitive** of degree n if it is transitive and preserves no nontrivial partition of $\{1, \dots, n\}$, where nontrivial partition means a partition that is not a partition into singleton sets or partition into one set. GAP contains a database with all primitive groups of degree < 4096 .

Two exercises from Peter Cameron's book⁷:

1. There is no sharply 4-transitive group of degree seven or nine.
2. Primitive groups of degree eight are 2-transitive.

⁷Permutation groups.

Let us construct the representation ρ of Alt_4 given by

$$(12)(34) \mapsto \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 0 & -1 \end{pmatrix}, \quad (123) \mapsto \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix}.$$

Representation theory

We use the function `GroupHomomorphismByImages`.

```
gap> A4 := AlternatingGroup(4);;
gap> a := [[0,1,-1],[1,0,-1],[0,0,-1]];
gap> b := [[0,0,-1],[0,1,-1],[1,0,-1]];
gap> rho := GroupHomomorphismByImages(A4,\
> [ (1,2)(3,4), (1,2,3) ], [ a, b ]);
gap> IsGroupHomomorphism(rho);
true
```

This is indeed a **faithful representation** of Alt_4 .

```
gap> IsTrivial(Kernel(rho));
true
```

Representation theory

Just to see how it works, let us compute $\rho_{(132)}$, the image of (132) under ρ . Display.

```
gap> Display(Image(rho, (1,3,2)));  
[ [ -1, 0, 1 ],  
  [ -1, 1, 0 ],  
  [ -1, 0, 0 ] ]
```

Now we construct the character χ of ρ . We also check that ρ is **irreducible** since

$$\langle \chi, \chi \rangle = \frac{1}{12} \sum_{g \in \text{Alt}_4} \chi(g) \chi(g^{-1}) = 1.$$

```
gap> chi := x->TraceMat(x^rho);;  
gap> 1/Order(A4)*\  
> Sum(List(A4, x->chi(x)*chi(x^(-1))));  
1
```

A problem of Brauer

Brauer⁸ asked what algebras are **group algebras**. This question might be impossible to answer. However, we can play with some particular examples.

⁸Lectures on Modern Mathematics, Vol I, 133–175, 1963.

A problem of Brauer

Is $\mathbb{C}^5 \times M_5(\mathbb{C})$ a (complex) **group algebra**? No. We will show that the groups algebras of groups of order 30 are only

$$\mathbb{C}^{10} \times M_2(\mathbb{C})^5, \quad \mathbb{C}^6 \times M_2(\mathbb{C})^6, \quad \mathbb{C}^2 \times M_2(\mathbb{C})^7, \quad \mathbb{C}^{30}.$$

To prove our claim, we can compute the **degrees** of the irreducible characters using `CharacterDegrees`. There are four groups of order 30 and none of them has a group algebra isomorphic to $\mathbb{C}^5 \times M_5(\mathbb{C})$.

```
gap> n := 30;;  
gap> for G in AllGroups(Size, n) do  
> Print(CharacterDegrees(G), "\n");  
> od;  
[ [ 1, 10 ], [ 2, 5 ] ]  
[ [ 1, 6 ], [ 2, 6 ] ]  
[ [ 1, 2 ], [ 2, 7 ] ]  
[ [ 1, 30 ] ]
```

Constructing irreducible representations

How can we **construct irreducible representations** of a given group?
This can be done with the package `Repsn`, written by Vahid Dabaghian.

Constructing irreducible representations

Let us construct the **irreducible representations** of Sym_3 . The **irreducible characters** of a finite group can be constructed with `Irr`:

```
gap> S3 := SymmetricGroup(3);;
gap> l := Irr(S3);
[ Character( CharacterTable( Sym( [ 1 .. 3 ] ) ),
  [ 1, -1, 1 ] ),
  Character( CharacterTable( Sym( [ 1 .. 3 ] ) ),
  [ 2, 0, -1 ] ),
  Character( CharacterTable( Sym( [ 1 .. 3 ] ) ),
  [ 1, 1, 1 ] ) ]
```

Constructing irreducible representations

To construct irreducible representations we need to load the package `repsn`:

```
gap> LoadPackage("repsn");
```

The package contains `IrreducibleAffordingRepresentation`. This function produces **irreducible representations from irreducible characters**.

Since we are working with Sym_3 , we will only need to consider the character of degree two. We will produce the faithful representation $\text{Sym}_3 \rightarrow \mathbf{GL}(2, \mathbb{C})$ given by

$$(123) \mapsto \begin{pmatrix} \omega^2 & 0 \\ 0 & \omega \end{pmatrix}, \quad (12) \mapsto \begin{pmatrix} 0 & \omega \\ \omega^2 & 0 \end{pmatrix},$$

where ω is a primitive cubic root of one.

Constructing irreducible representations

Here is the code:

```
gap> f := IrreducibleAffordingRepresentation(1[2]);  
[ (1,2,3), (1,2) ] ->  
[ [ [ E(3)^2, 0 ], [ 0, E(3) ] ],  
  [ [ 0, E(3) ], [ E(3)^2, 0 ] ] ]  
gap> Image(f, (1,2,3));  
[ [ E(3)^2, 0 ], [ 0, E(3) ] ]  
gap> Display(Image(f, (1,2,3)));  
[ [ E(3)^2, 0 ],  
  [ 0, E(3) ] ]  
gap> Display(Image(f, (1,2)));  
[ [ 0, E(3) ],  
  [ E(3)^2, 0 ] ]
```

An exercise on irreducible representations

Construct the irreducible representations of the groups \mathbb{D}_8 , $\mathbf{SL}_2(3)$, Alt_4 , Sym_4 and Alt_5 .

The McKay conjecture

For a finite group G and a prime p such that p divides $|G|$ one defines $\text{Irr}_{p'}(G) = \{\chi \in \text{Irr}(G) : p \nmid \chi(1)\}$.

Conjecture (McKay, 1970)

If $P \in \text{Syl}_p(G)$, then $|\text{Irr}_{p'}(G)| = |\text{Irr}_{p'}(N_G(P))|$.

It is believed that the McKay conjecture is true. Recently Malle and Späth proved the conjecture is true for $p = 2$.

The McKay conjecture

We write a naive function that checks the conjecture for a given group.

```
gap> McKay := function(G)
> local N, n, m, p;
> for p in PrimeDivisors(Order(G)) do
> N := Normalizer(G, SylowSubgroup(G, p));
> n := Number(Irr(G), x->Degree(x) mod p <> 0);
> m := Number(Irr(N), x->Degree(x) mod p <> 0);
> if not n = m then
> return false;
> fi;
> od;
> return true;
> end;
function( G ) ... end
```

The McKay conjecture

With this function is now easy to check the McKay conjecture in several small examples.

```
gap> McKay(SL(2,3));  
true  
gap> McKay(MathieuGroup(11));  
true  
gap> McKay(SuzukiGroup(8));  
true  
gap> McKay(PSL(2,7));  
true
```

The McKay conjecture

How can we check the conjecture say for other [sporadic simple groups](#)?

The package `AtlasRep` provides a nice interface between GAP and databases such as the [Atlas of Group Representations](#). The package contains information of simple groups such as generators, matrix and permutation representations, maximal subgroups, conjugacy classes.

The McKay conjecture

With `AtlasGroup` we create sporadic simple groups. Let us check McKay conjecture for the first [Janko group](#), a non-abelian simple group of order 175560.

```
gap> J1 := AtlasGroup("J1");;  
gap> Order(J1);  
175560  
gap> McKay(J1);  
true
```

The Isaacs–Navarro conjecture

For $k \in \mathbb{Z}$ and a finite group G let

$$M_k(G) = |\{\chi \in \text{Irr}_{p'}(G) : \chi(1) \equiv \pm k \pmod{p}\}|.$$

Conjecture (Isaacs–Navarro, 2002)

If $P \in \text{Syl}_p(G)$, then $M_k(G) = M_k(N_G(P))$.

The Isaacs–Navarro conjecture

Here we have a function that checks the Isaacs–Navarro conjecture:

```
gap> IsaacsNavarro := function(G, k, p)
> local mG, mN, N;
> N := Normalizer(G, SylowSubgroup(G, p));
> mG := Number(Filtered(Irr(G), x->Degree(x)\
> mod p <> 0), x->Degree(x) mod p in [-k,k] mod p);
> mN := Number(Filtered(Irr(N), x->Degree(x)\
> mod p <> 0), x->Degree(x) mod p in [-k,k] mod p);
> if mG = mN then
> return mG;
> else
> return false;
> fi;
> end;
function( G, k, p ) ... end
```

The Isaacs–Navarro conjecture

Let us check that the Isaacs–Navarro conjecture is true for the group $\mathbf{SL}_2(3)$. We only need to check the conjecture for $k \in \{1, 2\}$ and $p \in \{2, 3\}$.

```
gap> IsaacsNavarro(SL(2,3), 1, 2);
```

```
4
```

```
gap> IsaacsNavarro(SL(2,3), 1, 3);
```

```
6
```

```
gap> IsaacsNavarro(SL(2,3), 2, 2);
```

```
0
```

```
gap> IsaacsNavarro(SL(2,3), 2, 3);
```

```
6
```

The Ore conjecture

In 1951 Ore conjectured that every element of a finite non-abelian simple group is a commutator. In 2010 Liebeck, O'Brien, Shalev and Tiep proved the Ore conjecture.

In 1896 Frobenius proved that an element g of a finite group is a commutator if and only if

$$\sum \frac{\chi(g)}{\chi(1)} \neq 0,$$

where the sum is over the set of all irreducible characters of G .

The Ore conjecture

We write a function that for a given element g of a group G , returns the sum used by Frobenius to test whether the element g is a commutator of G .

```
gap> IsCommutator := function(group, g)
> local f, s;
> s := 0;
> for f in Irr(group) do
> s := s+g^f/Degree(f);
> od;
> return s;
> end;
function( group, g ) ... end
```

The Ore conjecture

We verify the conjecture for several **small** non-abelian simple groups.

```
gap> G := AlternatingGroup(5);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
gap> G := AlternatingGroup(6);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
gap> G := PSL(2,7);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
gap> G := PSL(2,8);;
gap> ForAll(G, g->IsCommutator(G, g) <> 0);
true
```

The Ore conjecture

The calculations needed only depend on the character table of the group, so we can make things better.

```
gap> Ore := function(ct)
> local f, s, x;
> for x in [1..NrConjugacyClasses(ct)] do
> s := 0;
> for f in Irr(ct) do
> s := s+f[x]/Degree(f);
> od;
> if s = 0 then
> return false;
> fi;
> od;
> return true;
> end;
function( ct ) ... end
```

The Ore conjecture

Now it is easy to verify Ore's conjecture for several simple groups!

```
gap> Ore(CharacterTable("J1"));  
true  
gap> Ore(CharacterTable("Co1"));  
true  
gap> Ore(CharacterTable("M24"));  
true  
gap> Ore(CharacterTable("Suz"));  
true  
gap> Ore(CharacterTable("HS"));  
true  
gap> Ore(CharacterTable("B"));  
true  
gap> Ore(CharacterTable("M"));  
true
```

Non-commutative ring theory

A ring R is said to be **Jacobson radical** if

$$R = \{x \in R : \text{there exists } y \in R \text{ such that } x + y + xy = 0\}.$$

To check whether a finite ring is Jacobson radical:

```
gap> IsJacobsonRadical := function(ring)
> local x, rad;
> rad := [];
> for x in ring do
> if not First(ring,\
> y->x+y+x*y=Zero(ring)) = fail then
> Add(rad, x);
> fi;
> od;
> return Size(ring)=Size(rad);
> end;
function( ring ) ... end
```

Non-commutative ring theory

The ring $\mathbb{Z}/3$ of integers mod 3 is not Jacobson radical. The subring $\{0,2\}$ of $\mathbb{Z}/4$ is Jacobson radical.

```
gap> IsJacobsonRadical(Integers mod 3);  
false  
gap> ring := Integers mod 4;;  
gap> subring := Subring(ring, [ZmodnZObj(0,4),\  
> ZmodnZObj(2,4)]);;  
gap> Elements(subring);  
[ ZmodnZObj( 0, 4 ), ZmodnZObj( 2, 4 ) ]  
gap> IsJacobsonRadical(subring);  
true
```