Worksheet on Homomorphisms in GAP

Alexander Hulpke

November 20, 2018

Exercise 1 (Form some homomorphisms to quotients and compare their images)

a) *Let* `G:=SmallGroup(60,10)`. *Construct the homomorphism* $\varphi\colon G \to G/G'$.

b) *Let* $G = S_5$. *Construct the homomorphism* $\varphi\colon G \to G/G'$.

c) *Let* $G = SL_2(5)$. *Construct the homomorphism* $\varphi\colon G \to G/G'$.

d) *Can you explain the kind of image group, and why the third homomorphism looks so different.*

□

Exercise 2 (A homomorphism given by a rule) *Let* G *be the stabilizer in* $GL_4(3)$ *of the subspace spanned by the first two basis vectors:*

```
gap> sub:=IdentityMat(4,GF(3)){[1,2]};
[ [ Z(3)^0,0*Z(3),0*Z(3),0*Z(3) ], [ 0*Z(3),Z(3)^0,0*Z(3),0*Z(3) ] ]
gap> G:=Stabilizer(GL(4,3),sub,OnSubspacesByCanonicalBasis);
<matrix group of size 186624 with 7 generators>
```

Construct a homomorphism on $GL_2(3)$ *given by the action on the factor space.* □

Exercise 3 (Create a presentation and transfer to another free group)
Construct $S_4 = \langle e := (1,2,3,4), f := (1,2)\rangle$ *as a finitely presented group on these two generators (named* `e` *and* `f`*).*
Hint: *Use* `IsomorphismFpGroupByGenerators` *to get a presentation, then make a new free group with the generators named as you want and transfer the relators to the new group. Use* `FreeGroupOfFpGroup` *and* `RelatorsOfFpGroup` *to translate relators to the free group with the desired generator names.* □

Exercise 4 (Stabilizer of a set of sets by changing permutation representation)
The Fano plane *is a combinatorial configuration consisting of 7 points and 7 three-element subsets of these points:*

$$\{\{1,2,3\},\{1,4,5\},\{1,6,7\},\{2,4,6\},\{2,5,7\},\{3,4,7\},\{3,5,6\}\}$$

The automorphism group of this configuration is the subgroup of S_7 that fixes these sets. To find it:

a) *Construct all 35 3-element sets out of 7 points and construct the homomorphism for the action of S_7 on these sets.*

b) *Represent the fano plane by a 7-of-35 set of points and calculate the stabilizer of this set under the image of the action in a)*

c) *Take the pre-image of this stabilizer.*

□

Exercise 5 (Automorphisms induced by normalizer in S_n)
Let $G = \langle (1,2),(3,4)(5,6),(3,6)(4,5),(7,8),(3,4)(9,10),(3,9)(4,10)\rangle$. What is the number of automorphisms of G that are induced by $N_{S_{10}}(G)$?
Hint: *Calculate the normalizer and use* `ConjugatorAutomorphism` *on its generators. Bonus: Can you find the same result from calculating the automorphism group* without *calculating the normalizer?* □

Exercise 6 (Outer automorphism of S_6) *Construct an outer automorphism of S_6.* □

Exercise 7 (Smallest faithful permutation degree) *Let* `G:=SmallGroup(48,48);`*. What is the smallest degree n, such that G is isomorphic to a subgroup of S_n?* □

Exercise 8 (Construct an (external) semidirect product) *Construct S_4 as semidirect product $S_3 \ltimes C_2^2$* □

Exercise 9 (Decompose a group formally as (internal) semidirect product)
Let `G:=PerfectGroup(IsPermGroup,960,1);` *and N its solvable radical (the largest solvable normal subgroup,* `RadicalGroup`*). Find a complement C to N in G, using the operation* `ComplementClassesRepresentatives`*. Form (directly, without using* `IsomorphismGroups`*) an isomorphism between G/N and C and use it to write elements of G as pairs in $C \ltimes N$. Build an isomorphism between G and an abstract semidirect product of N with G.* □

Exercise 10 (Quotients of a particular type — generating direct products)
Determine the largest k, such that A_6^k can be generated by 2 elements. Construct such a 2-element generating set. **Hint:**

a) `GQuotients` *finds Quotients of a given isomorphism group. What happens if you apply it to the free group?*

b) *What happens if you intersect the kernels of multiple quotients with the same simple image?*

c) *Combine the homomorphisms to one homomorphism into the direct product of the images.* □

Exercise 11 (Forming larger and larger quotients of finitely presented groups) *Let $G = \langle x, y, u, v \mid x^2, y^3, u^{10}v^2uvuv^2, xy/u, xy^{-1}/v \rangle$. Construct a large quotient of G.*
Hint: *G has order $2^{21}3^4 5^2 13^2$.*

a) *Use* `IsomorphismSimplifiedFpGroup` *to eliminate two generators.*

b) *Find subgroups of low index (as far as plausible, at least 30) and intersect. Take the homomorphism φ that is used to define this intersection.*

c) *Find a subgroup of small index (*`MaximalSubgroupClassReps`, `LowLayerSubgroups`*) of the image of φ such that its abelianization is different than the one of its pre-image under φ.*

d) *Repeat, if needed.* □

Exercise 12 (Constructing and inducing representations) *Construct the* reduced permutation action *of A_5 over $GF(3)$ (That is the action on the vectors whose coefficients sum to one.) Then induce this representation to $PSL_2(9)$.*
Hint: *You need to construct the map*

$$g \mapsto \left(g^{\psi}; \widetilde{g_{1^{g^{-1}}}}^{\varphi}, \ldots, \widetilde{g_{n^{g^{-1}}}}^{\varphi}\right)$$

where $\widetilde{g_j} \in T$ is defined by by $r_j g = \widetilde{g_j} r_{j^g}$ for coset representatives r_j. □

Answers

1

Exercise 1, page 1

The particular `NaturalHomomorphismByNormalSubgroup` also can be called as `MaximalAbelianQuotient`.

```
gap> G:=SmallGroup(60,10);;
gap> NaturalHomomorphismByNormalSubgroup(G,DerivedSubgroup(G));
[ f1, f2, f3, f4 ] -> [ f1, f2, f3, <identity> of ... ]
gap> G:=SymmetricGroup(5);;
gap> MaximalAbelianQuotient(G);
[ (1,2,3,4,5), (1,2) ] -> [ <identity> of ..., f1 ]
gap> MaximalAbelianQuotient(SL(2,5));
CompositionMapping( [ (2,5,4,3)(6,11,16,21 [...] ] -> [ (), () ],
  <action isomorphism> )
```

The image is usually a Pc group, as these are the most efficient to work with (and easiest to build the factor group). In the third case an action on vectors is taken first, as generically GAP (without package) cannot decompose in matrix groups. (In the last case it is a permutation group, as it was not possible to generate a trivial Pc group.)

Exercise 2, page 1

We need to cut out the submatrix at positions $(3, 4)$. This is easiest done by a function.

```
gap> fct:=mat->mat{[3,4]}{[3,4]};;
gap> hom:=GroupHomomorphismByFunction(G,GL(2,3),fct);
MappingByFunction( <matrix group of size 186624 with
7 generators>, GL(2,3), function( mat ) ... end )
```

Exercise 3, page 1

First generate the group and a presentation in the desired generators:

```
gap> s4:=Group((1,2,3,4),(1,2));;
gap> pres:=IsomorphismFpGroupByGenerators(s4,GeneratorsOfGroup(s4));
[ (1,2,3,4), (1,2) ] -> [ F1, F2 ]
gap> fp1:=Range(pres);
<fp group of size 24 on the generators [ F1, F2 ]>
gap> RelatorsOfFpGroup(fp1);
[ F2^2, F1^4, (F2*F1^-1)^3 ]
```

Now create another free group with desired generator names, map the relators, and create the corresponding quotient.

```
gap> f:=FreeGroup("e","f");;
gap> fiso:=GroupHomomorphismByImages(FreeGroupOfFpGroup(fp1),f,
> FreeGeneratorsOfFpGroup(fp1),GeneratorsOfGroup(f));
[ F1, F2 ] -> [ e, f ]
gap> newrels:=ImagesSet(fiso,RelatorsOfFpGroup(fp1));
[ f^2, e^4, (f*e^-1)^3 ]
gap> new:=f/newrels;
<fp group on the generators [ e, f ]>
gap> Size(new);
24
```

(Note: Instead of the isomorphism between the free groups, one could have used `MappedWord`.)

Exercise 4, page 2

```
gap> fano:=[[1,2,3],[1,4,5],[1,6,7],[2,4,6],[2,5,7],[3,4,7],[3,5,6]];;
gap> sets:=Combinations([1..7],3);;
gap> Length(sets);
35
gap> act:=ActionHomomorphism(s7,sets,OnSets,"surjective");
<action epimorphism>
gap> fans:=List(fano,x->Position(sets,x));
[ 1, 10, 15, 21, 24, 28, 29 ]
gap> sub:=Stabilizer(Image(act),fans,OnSets);
<permutation group of size 168 with 4 generators>
gap> au:=PreImage(act,sub);
Group([ (1,2)(5,6), (1,2,3)(5,6,7), (1,2,3)(4,7,6), (1,4,7,6,2,5,3) ])
```

Exercise 5, page 2

If we are permitted to calculate the normalizer, we can do so directly, determining for each normalizer generator the corresponding automorphism.

```
gap> G:=Group((1,2),(3,4)(5,6),(3,6)(4,5),(7,8),(3,4)(9,10),(3,9)(4,10));;
gap> S10:=SymmetricGroup(10);;
gap> n:=Normalizer(S10,G);
Group([ (1,7)(2,8), (1,8,2,7), (3,9)(4,10), (3,9,5,4,10,6) ])
gap> auts:=List(GeneratorsOfGroup(n),x->ConjugatorAutomorphism(G,x));
[ ^(1,7)(2,8), ^(1,8,2,7), ^(3,9)(4,10), ^(1,2)(3,10,5)(4,9,6)(7,8) ]
```

The group we want is generated by these automorphisms

```
gap> au:=AutomorphismGroup(G);
<group of size 576 with 4 generators>
gap> sub:=Subgroup(au,auts);;
gap> Size(sub);
48
```

Why did we bother to create the group as subgroup of the automorphism group? This way GAP knows that the group is generated by group automorphisms, and uses better ways to find the order. Alternatively, we would have to tell this fact to GAP:

```
gap> sub:=Group(auts);
<group with 4 generators>
gap> SetIsGroupOfAutomorphismsFiniteGroup(sub,true);
gap> Size(sub);
48
```

For the bonus part, we find the subgroup of the automorphism group that stabilizes the set of point stabilizers, collected by orbit lengths:

```
gap> o:=Orbits(G,MovedPoints(G));
[ [ 1, 2 ], [ 3, 4, 6, 9, 5, 10 ], [ 7, 8 ] ]
gap> oo:=[Union(o[1],o[3]),o[2]];
[ [ 1, 2, 7, 8 ], [ 3, 4, 6, 9, 5, 10 ] ]
gap> stabs:=List(o,x->List(x,p->Stabilizer(G,p)));;
gap> stabs:=List(stabs,Set);
[ [ Group([ (3,5)(4,6), (3,4)(5,6), (3,10,5)(4,9,6), (7,8) ]),
    Group([ (3,5)(4,6), (3,4) (5,6), (3,10,5)(4,9,6), (1,2) ]) ],
  [ Group([ (5,10)(6,9), (5,6)(9,10), (1,2), (7,8) ]),
    Group([ (3,9)(4,10), (3,4) (9,10), (1,2), (7,8) ]),
    Group([ (3,6)(4,5), (3,4)(5,6), (1,2), (7,8) ]) ] ]
```

Now we make a function that has automorphisms act on sets of lists of sets of subgroups, and take the stabilizer.

```
gap> act:=function(list,aut)
> return List(list,s->Set(List(s,g->Image(au,g))));
> end;;
gap> sub:=Stabilizer(au,stabs,act);
<group of size 48 with 5 generators>
```

Exercise 6, page 2

One such automorphism must be amongst the generators of the automorphism group:

```
gap> au:=AutomorphismGroup(SymmetricGroup(6));
<group with 3 generators>
gap> First(GeneratorsOfGroup(au),x->not IsInnerAutomorphism(x));
[ (5,6), (1,2,3,4,5) ] -> [ (1,2)(3,5)(4,6), (1,2,3,4,5) ]
```

Exercise 7, page 2

The type conversion does not help:

```
gap> NrMovedPoints(Range(IsomorphismPermGroup(G)));
48
```

The smart-alec solution is

```
gap> MinimalFaithfulPermutationDegree(G);
6
gap> IsomorphicSubgroups(SymmetricGroup(6),G);
[ [ f1*f2*f3*f4, f1*f4*f5 ] -> [ (3,4,5,6), (1,2)(3,4) ],
  [ f1*f2*f3*f4, f1*f4*f5 ] -> [ (2,5,4,6), (1,2)(3,4) ] ]
```

(So there are actually two such ways, a transitive one and an intransitive one.)

Without the slick functions, one has to look at sets of subgroups in the hope to find a few of small index, such that their cores intersect trivially. We try for different numbers k of orbits.

```
gap> u:=List(ConjugacyClassesSubgroups(G),Representative);;
gap> tup:=Filtered(Combinations(u,1),
  x->IsTrivial(Intersection(List(x,y->Core(G,y)))));;
gap> Length(tup);
16
gap> Set(List(tup,x->Sum(List(x,y->Index(G,y)))));
[ 6, 8, 12, 16, 24, 48 ]
gap> tup:=Filtered(Combinations(u,2),
  x->IsTrivial(Intersection(List(x,y->Core(G,y)))));;
gap> Length(tup);
434
gap> Set(List(tup,x->Sum(List(x,y->Index(G,y)))));
[ 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 24, 25,
26, 27, 28, 30, 32, 36, 40, 48, 49, 50, 51, 52, 54, 56, 60, 64, 72 ]
```

Clearly we only need to try k subgroups if the degree might be $\geq 2k$, that's why we stop here. Again we can note that there is an option with one orbit, and one with two orbits.

Exercise 8, page 2

We first need to find the action of S_3 on C_2^2, which we write as matrices. Note that the matrix action is faithful, so we could use the special version of `SemidirectProduct` for matrix groups:

```
gap> m1:=[[0,1],[1,0]]*One(GF(2)); m2:=[[1,1],[1,0]]*One(GF(2));
[ [ 0*Z(2), Z(2)^0 ], [ Z(2)^0,0*Z(2) ] ]
[ [ Z(2)^0, Z(2)^0 ], [ Z(2)^0,0*Z(2) ] ]
gap> sdp:=SemidirectProduct(Group(m1,m2),GF(2)^2);
<matrix group of size 24 with 3 generators>
```

However now, lets do it the hard way by writing out the details. First we need both groups.

```
gap> s3:=SymmetricGroup(3);
Sym( [ 1 .. 3 ] )
gap> v4:=AbelianGroup([2,2]);
<pc group of size 4 with 2 generators>
```

Next, we construct for each matrix automorphisms of the abelian group. For this we need to have generators that treat it like a vector space. The easiest is to use a polycyclic generating set, `Pcgs`. We then reado off the coefficients from the matrix entries.

```
gap> pcgs:=Pcgs(v4);
Pcgs([ f1, f2 ])
gap> aut1:=GroupHomomorphismByImages(v4,v4,pcgs,[pcgs[2],pcgs[1]]);
Pcgs([ f1, f2 ]) -> [ f2, f1 ]
gap> aut2:=GroupHomomorphismByImages(v4,v4,pcgs,
> [PcElementByExponents(pcgs,m2[1]),PcElementByExponents(pcgs,m2[2])]);
Pcgs([ f1, f2 ]) -> [ f1*f2, f1 ]
```

Note how in the second example we acturally read off coefficients directly from the matrix – that is how one would do it inside a general function.

Now we form a group from the automorphisms. (As we do not calculate in this group, we do not bother to tellit that it is a group of automorphisms.) Also construct a homomorphism from S_3 to this group.

```
gap> aut:=Group(aut1,aut2);
<group with 2 generators>
gap> hom:=GroupHomomorphismByImages(s3,aut,[(1,2),(1,2,3)],[aut1,aut2]);
[ (1,2), (1,2,3) ] ->
[ Pcgs([ f1, f2 ]) -> [ f2, f1 ], Pcgs([ f1, f2 ]) -> [ f1*f2, f1 ] ]
gap> sdp:=SemidirectProduct(s3,hom,v4);
<pc group of size 24 with 4 generators>
```

Note that GAP automatically decided on representing the product as a polycyclic group itself, as this is the most effiicent way for solvable groups. We can see that the pc presentation actually reflects the semidirect product structure.

```
gap> Image(Embedding(sdp,1));
Group([ f1, f2 ])
gap> Image(Embedding(sdp,2));
Group([ f3, f4 ])
```

Exercise 9, page 2

Construct the groups and get one complement.

```
gap> G:=PerfectGroup(IsPermGroup,960,1);;
gap> N:=RadicalGroup(G);;
gap> c:=ComplementClassesRepresentatives(G,N);
[ Group([ (2,14,9,13,15)(3,4,6,7,16)(5,11,12,10,8), (2,6,4,13,10)(3,11,5,7,15)
  (8,9,14,12,16) ]), Group([ (2,14,9,13,15 [...]
gap> c:=c[1];
```

To construct the isomorphism, we take the natural homomorphism $G \to G/N$ and restrict it to C. The isomorphism is the inverse (and we check that it is an isomorphism)

```
gap> nat:=NaturalHomomorphismByNormalSubgroup(G,N);
[ (2,4)(3,5)(7,15)(8,1[...] ] -> [ (1,2)(4,5), (1,3,4), (), (), (), () ]
gap> res:=RestrictedMapping(nat,c);
[ (2,14,9,13,15)(3,4,6,7,1  [...] ] -> [ (1,2,3,4,5), (1,3,2,4,5) ]
gap> iso:=InverseGeneralMapping(res);
[ (1,2,3,4,5), (1,3,2,4,5) ] -> [ (2,14,9,13,15)(3,4,6,7,16)(5,11,12,10,8),
  (2,6,4,13,10)(3,11,5,7,15)(8,9,14,12,16) ]
gap> IsBijective(iso);
true
```

To decompose, we get the C-part by mapping under the natural homomorphism and then into C, the N-part is obtained by dividing off the C-part. (Left or right – that is your choice, it gives isomorphic semidirect products.)

```
gap> cpart:=nat*iso;
[ (2,4)(3,5)(7,15)(8,14)(10,13)(12,16), [...] ] ->
[ (2,4)(3,5)(7,15)(8,14)(10,13)(12,16), [...], (), (), (), () ]
gap> decomp:=function(elm)
> local cp;
>   cp:=Image(cpart,elm);
>   return [cp,LeftQuotient(cp,elm)];
```

```
> end;;
gap> r:=Random(G);
(1,14,3)(2,4,7)(5,9,8)(6,10,12)(11,13,16)
gap> d:=decomp(r);
[ (2,5,10)(3,14,15)(4,16,8)(6,9,11)(7,12,13),
  (1,14)(2,12)(3,15)(4,5)(6,13)(7,16)(8,11)(9,10) ]
gap> d[1] in c;d[2] in N;
true
true
gap> d[1]*d[2];
(1,14,3)(2,4,7)(5,9,8)(6,10,12)(11,13,16)
```

Next construct an abstract semidirect product. For that we need to construct the automorphisms of N induced by the generators of c and construct a homomorphism form c to the group of these induced automorphisms.

With that, we can construct the semidirect product.

```
gap> auts:=List(GeneratorsOfGroup(c),x->ConjugatorIsomorphism(N,x));
[ ^(2,14,9,13,15)(3,4,6,7,16)(5,11,12,10,8),
  ^(2,6,4,13,10)(3,11,5,7,15)(8,9,14,12,16) ]
gap> autgp:=Group(auts);;
gap> map:=GroupHomomorphismByImages(c,autgp,GeneratorsOfGroup(c),auts);;
gap> sdp:=SemidirectProduct(c,map,N);
<permutation group with 6 generators>
gap> sdp=G;
false
```

We see that (of course) it does not have the same elements, as G. There now are two ways to construct an isomorphism. The first version is to give it on generators of C and N separately. That doesn't really need the decomposition. We simply use the `Embedding` command:

```
gap> cimgs:=List(GeneratorsOfGroup(c),x->Image(Embedding(sdp,1),x));
[ (1,13,8,12,14)(2,3,5,6,15)(4,10,11,9,7)(17,29,24,28,30) [...] ]
gap> nimgs:=List(GeneratorsOfGroup(N),x->Image(Embedding(sdp,2),x));
[ (16,17)(18,23)(19,24)(20,25)(21,22)(26,30)(27,29)(28,31),
  (16,18)(17,23)(19,28)(20,21)(22,25)(24,31)(26,27)(29,30), [...] ]

gap> iso:=GroupHomomorphismByImages(G,sdp,Concatenation(GeneratorsOfGroup(c),
> GeneratorsOfGroup(N)),Concatenation(cimgs,nimgs));
[ (2,14,9,13,15)(3,4,6,7,16)(5,11,12,10,8), (2,6,4,1[...]
```

Alternatively, we could take an existing generating set of G and use our decomposition function.

```
gap> gens:=SmallGeneratingSet(G);
[ (1,9,2,5,7)(3,8,15,4,14)(6,12,11,10,13), (1,14,16,13,11 [...]  ]
gap> decomps:=List(gens,decomp);
[ [(2,12,6,5,15)(3,13,11,14,16)(4,10,8,7,9),(1,9)(2,4)(3,16)(5,12[...] ]
gap> imgs:=List(decomps,
> x->Image(Embedding(sdp,1),x[1])*Image(Embedding(sdp,2),x[2]));
[ (1,11,5,4,14)(2,12,10,13,15)(3,9,7,6,8)(16,24,17,20 [...] ]
gap> iso2:=GroupHomomorphismByImages(G,sdp,gens,imgs);
[ (1,9,2,5,7)(3,8,15,4,14)(6,12,11 [...] ] -> [ (1,11, [...] ]
```

Exercise 10, page 2

```
gap> f:=FreeGroup(2);;
gap> q:=GQuotients(f,SimpleGroup("A6"));;
```

finds 53 quotients with different kernels. Their intersection will have quotient A_6^{53}, as the direct product has only the obvious normal subgroups. (This is different if the group is not simple!) If A_6^{54} could be generated by two elements, there would have to be another quuotient.

To construct an explicit generator pair, we need to construct the map onto A_6^{53}:

```
gap> a6:=AlternatingGroup(6);;
gap> l:=ListWithIdenticalEntries(Length(q),a6);;
gap> dir:=DirectProduct(l);
<permutation group of size 30481[...] with 106 generators>
```

Next we take for each generator of the free group the images under each of the quotients and embed into the corresponding copy of the direct product, and take the product. (Note the slick use of `f.(g)` to access a generator with a variable number.)

```
gap> gens:=List([1,2],g->Product([1..Length(q)],i->Image(Embedding(dir,i),
> Image(q[i],f.(g)))));
[ (2,3)(4,5)(9,10)(11,12)(14,16)(17,18)(21,22,23) [...]
```

Finally we verify.

```
gap> Size(Group(gens))=Size(dir);
true
```

Exercise 11, page 3

We start constructing the group on two generators and find a quotient from `LowIndexSubgroups`.

```
gap> f:=FreeGroup("x","y","u","v");;
gap> rels:=ParseRelators(f,"x2,y3,u10v2uvuv2,xyU,xYV");
[ x^2, y^3, u^10*v*(v*u)^2*v^2, x*y*u^-1, x*y^-1*v^-1 ]
gap> g:=f/rels;
<fp group on the generators [ x, y, u, v ]>
gap> IsomorphismSimplifiedFpGroup(g);
[ x, y, u, v ] -> [ u*y^-1, y, u, u*y^-2 ]
gap> g:=Image(last);
<fp group on the generators [ y, u ]>
```

Next we construct the homomorphism arising from the intersection of low index subgroups

```
gap> l:=LowIndexSubgroups(g,30);;
gap> List(l,IndexInWholeGroup);
[ 1, 13, 26, 13, 26, 26, 26, 26, 26, 26 ]
gap> k:=Intersection(l);
Group(<fp, no generators known>)
gap> q:=DefiningQuotientHomomorphism(k);;
gap> p:=Image(q);; Collected(Factors(Size(p)));
[ [ 2, 19 ], [ 3, 4 ], [ 5, 2 ], [ 13, 2 ] ]
```

Next we find a maximal subgroup for which the abelianization differs and take the preimage

```
gap> m:=MaximalSubgroupClassReps(p);
[...]
gap> m:=ShallowCopy(m);; # as m is immutable
gap> SortBy(m,x->Index(p,x));
gap> List(m,x->Index(p,x));
[ 13, 13, 26, 65, 65, 144, 234, 300, 325, 4096, 4096 ]

gap> sub:=First(m,x->AbelianInvariants(x)<>AbelianInvariants(PreImage(q,x)));;
gap> Index(p,sub);
4096
gap> AbelianInvariants(sub);
[  ]
gap> u:=PreImage(q,sub);
Group(<fp, no generators known>)
gap> AbelianInvariants(u);
[ 2 ]
```

In this case it is enough to take the derived subgroup. (Note: A slicker version is given by `LargerQuotientBySubgroupAbelianization`, which tries to avoid already known abelianization parts.)

```
gap> new:=DerivedSubgroup(u);;
gap> new:=LargerQuotientBySubgroupAbelianization(q,sub);; # alternative
gap> Index(g,new);
8192
```

As the index is comparatively large, we do not try `Intersection` in `g`, but embed by hand into a direct product 2).

```
gap> q2:=DefiningQuotientHomomorphism(new);;
gap> p2:=Image(q2);;
gap> Size(p2);
46006272
gap> dir:=DirectProduct(p,p2);;
gap> imgs:=List(GeneratorsOfGroup(g),x->Image(Embedding(dir,1),Image(q,x))*
> Image(Embedding(dir,2),Image(q2,x)));
[ (2,3,4)(5,6,7)(8,9,11)(10,13,17)(12,15,19)(14,18,20)(16,21,23)(22,24,2  [...]
gap> p3:=SubgroupNC(dir,imgs);
gap> Collected(Factors(Size(p3)));
[ [ 2, 20 ], [ 3, 4 ], [ 5, 2 ], [ 13, 2 ] ]
```

So get the remaining 2, we have to repeat using second maximals as well:

```
gap> m:=LowLayerSubgroups(p3,2);
gap> m:=ShallowCopy(m);;
gap> SortBy(m,x->Index(p3,x));
gap> sub:=First(m,x->AbelianInvariants(x)<>AbelianInvariants(PreImage(q3,x)));;
gap> Index(p3,sub);
3744
gap> AbelianInvariants(sub);
[ 3, 3, 4 ]
gap> AbelianInvariants(PreImage(q3,sub));
[ 3, 3, 8 ]
```

Now `LargerQuotientBySubgroupAbelianization` really helps, as it ignores the 3^2. We then do the same embedding into a direct product

```
gap> new3:=LargerQuotientBySubgroupAbelianization(q3,sub);
Group(<fp, no generators known>)
gap> Index(g,new3);
29952
gap> q4:=DefiningQuotientHomomorphism(new3);;
gap> p4:=Image(q4);
<permutation group of size 358848921600 with 2 generators>
gap> NrMovedPoints(p4);
29952
```

```
gap> Collected(Factors(Size(p4)));
[ [ 2, 20 ], [ 3, 4 ], [ 5, 2 ], [ 13, 2 ] ]
gap> dir:=DirectProduct(p3,p4);;
gap> imgs:=List(GeneratorsOfGroup(g),x->Image(Embedding(dir,1),Image(q3,x))*
> Image(Embedding(dir,2),Image(q4,x)));;
gap> p5:=Group(imgs);;
gap> Collected(Factors(Size(p5)));
[ [ 2, 21 ], [ 3, 4 ], [ 5, 2 ], [ 13, 2 ] ]
```

If we take the order of the group as given, we know that `p5` is a faithful permutation image.

Exercise 12, page 3

Construct a basis and the action. (The permutation orbit is slightly overkill...)

```
gap> a5:=AlternatingGroup(5);
gap> vec:=[1,-1,0,0,0]*One(GF(3));
[ Z(3)^0,Z(3),0*Z(3),0*Z(3),0*Z(3) ]
gap> o:=Orbit(Group((1,2,3,4,5)),vec,Permuted);
[ [ Z(3)^0,Z(3),0*Z(3),0*Z(3),0*Z(3) ], [ 0*Z(3),Z(3)^0,Z(3),0*Z(3),0*Z(3) ],
  [ 0*Z(3),0*Z(3),Z(3)^0,Z(3),0*Z(3) ], [ 0*Z(3),0*Z(3),0*Z(3),Z(3)^0,Z(3) ],
  [ Z(3),0*Z(3),0*Z(3),0*Z(3),Z(3)^0 ] ]
gap> o:=Filtered(TriangulizedMat(o),x->not IsZero(x));
[ [ Z(3)^0,0*Z(3),0*Z(3),0*Z(3),Z(3) ], [ 0*Z(3),Z(3)^0,0*Z(3),0*Z(3),Z(3) ],
  [ 0*Z(3),0*Z(3),Z(3)^0,0*Z(3),Z(3) ], [ 0*Z(3),0*Z(3),0*Z(3),Z(3)^0,Z(3) ] ]
gap> mats:=List(GeneratorsOfGroup(a5),
> g->List(o,v->SolutionMat(o,Permuted(v,g))));
[ [ [ Z(3),Z(3)^0,0*Z(3),0*Z(3) ], [ Z(3),0*Z(3),Z(3)^0,0*Z(3) ],
      [ Z(3),0*Z(3),0*Z(3),Z(3)^0 ], [ Z(3),0*Z(3),0*Z(3),0*Z(3) ] ],
  [ [ Z(3)^0,0*Z(3),Z(3),0*Z(3) ], [ 0*Z(3),Z(3)^0,Z(3),0*Z(3) ],
      [ 0*Z(3),0*Z(3),Z(3),Z(3)^0 ], [ 0*Z(3),0*Z(3),Z(3),0*Z(3) ] ] ]
gap> rep:=GroupHomomorphismByImages(a5,Group(mats),GeneratorsOfGroup(a5),mats);
[ (1,2,3,4,5), (3,4,5) ] -> [...]
```

Note: There is `IrreducibleRepresentations`, but that produces irrational entries.

To induce to $\mathrm{PSL}_2(9)$, we construct the group and find an embedding of A_5, as well as a transversal and action on cosets. We also make rep be defined on this subgroup.

```
gap> psl:=PSL(2,9);
Group([ (3,9,4,6)(5,10,8,7), (1,2,4)(5,6,8)(7,9,10) ])
gap> max:=Filtered(MaximalSubgroupClassReps(psl),x->Size(x)=60);
[ Group([ (1,10,5,2,3)(4,8,6,9,7), (1,10,7)(3,9,8)(4,5,6) ]),
  Group([ (1,10)(2,7)(3,6)(5,8), (2,4,3)(5,10,6)(7,8,9) ]) ]
gap> iso:=IsomorphismGroups(a5,max[1]);
```

```
[ (2,4)(3,5), (1,2,3) ] -> [ (1,2)(5,10)(6,9)(7,8), (2,5,8)(3,6,7)(4,10,9) ]
gap> t:=RightTransversal(psl,Image(iso));
RightTransversal(Group([ (3,9,4,6)(5,10,8,7), (1,2,4)(5,6,8)(7,9,10) ]),
  Group([ (1,10,5,2,3)(4,8,6,9,7), (1,10,7)(3,9,8)(4,5,6) ]))
gap> ca:=ActionHomomorphism(psl,t,OnRight); # documented abuse of notation
<action homomorphism>
gap> rep:=InverseGeneralMapping(iso)*rep;
[ (1,2)(5,10)(6,9)(7,8), (2,5,8)(3,6,7)(4,10,9) ] ->
[ [ [ Z(3)^0,0*Z(3),Z(3),0*Z(3) ], [ 0*Z(3),0*Z(3),Z(3),Z(3)^0 ],
      [ 0*Z(3),0*Z(3),Z(3),0*Z(3) ], [ 0*Z(3),Z(3)^0,Z(3),0*Z(3) ] ],
  [ [ 0*Z(3),Z(3)^0,0*Z(3),0*Z(3) ], [ 0*Z(3),0*Z(3),Z(3)^0,0*Z(3) ],
      [ Z(3)^0,0*Z(3),0*Z(3),0*Z(3) ], [ 0*Z(3),0*Z(3),0*Z(3),Z(3)^0 ] ] ]
```

The induction to $\mathrm{PSL}_2(9)$ is a subgroup of the wreath product $\mathrm{GL}_4(3) \wr S_6$. Construct it and the seven embeddings (Number 7 is the complement):

```
gap> wr:=WreathProduct(GL(4,3),SymmetricGroup(6));
<matrix group of size 14682417[...] with 4 generators>
gap> emb:=List([1..7],i->Embedding(wr,i));;
```

As the map looks somewhat complicated, it might be easier to have it be produced by a function, rather than a complicated `List` construct:

```
myimg:=function(elm)
local perm,img,i,j,r;
  perm:=Image(ca,elm);
  img:=Image(emb[7],perm);
  for i in [1..6] do
    j:=i/perm;
    r:=t[j]*elm/t[i];
    img:=img*Image(emb[i],Image(rep,r));
  od;
  return img;
end;

gap> ind:=GroupGeneralMappingByImagesNC(psl,wr,GeneratorsOfGroup(psl),
> List(GeneratorsOfGroup(psl),myimg));;
gap> IsSingleValued(ind);
true
```

We use GroupGeneralMappingByImagesNC to avoid expensive element tests in `wr`. The final `IsSingleValued` verifies that is is a homomorphism. Instead (without test) we could have used:

```
gap> ind:=GroupHomomorphismByFunction(psl,wr,myimg);;
gap> ind:=AsGroupGeneralMappingByImages(ind);;
```