TEACHING ABSTRACT ALGEBRA WITH GAP

Dr. Kevin Lee Anderson
Missouri Western State College
4525 Downs Drive
St. Joseph, Missouri, 64507
andersk@mwsc.edu

GAP stands for Groups, Algorithms, and Programming. GAP is a freeware package that can be found on the web at http://www.gap-system.org/. GAP was originally written for UNIX but as been ported to Windows and Mac systems. The program runs is a DOS window so it isn't as flashy as the typical Windows program, and takes a little work to get used to. This being said, GAP is a very powerful program for computational discrete algebra, with particular emphasis on group theory. What GAP lacks in "style" it more then makes up for in "power".

GAP is designed mainly for research in computational discrete algebra. My first introduction to GAP was in graduate school. Where I used it to investigate certain finite group presentations that arouse in my study of Coxeter-Petrie Complexes of Regular Affine Maps.

Many people use GAP in their research, including myself. It has built in programs and packages for working with finitely presented groups, finite fields, cohomology, and much more. GAP has a built in programming language, which is a little tricky to learn. One doesn't need to know how to program to use GAP effectively in the classroom though. The nice thing about GAP is it's easy to learn the basics and has many built in functions you and your students can use to explore Abstract Algebra. To get acquainted with GAP I highly recommend the on line tutorial http://www-history.mcs.st-and.ac.uk/~gap/Manual4/htm/tut/chapters.htm There is also a very nice search engine on the main page, which can be used to search the GAP manuals and forums. The complete GAP distribution is rather large, over 300MB, so downloading it with a 56k modem should be avoided at all costs. Being a "ported" program GAP doesn't have an automated setup program, but the instructions are straightforward and if you know how to unzip a file you pretty much know how to load gap.

In my Abstract Algebra class I use GAP as a "fancy calculator" and as a tool to help the students do calculations in order to generate conjectures. One thing I wanted to avoid was turning my Abstract Algebra class into a "GAP class." Anyone who's taught College Algebra with a TI calculator knows what I'm talking about. Your College Algebra class can easily turn into a class on how to use the TI calculator! Robbing your students of needed skills.

To avoid the above issues I kept in-class examples of GAP small and simple. More detailed examples were given in the form of worksheets outside of class. Again to avoid teaching GAP all the worksheets were straight forward using the built in functions of GAP.

There is another system similar to GAP called MAGMA. MAGMA is produced and licensed by the University of Sydney, Australia http://magma.maths.usyd.edu.au/magma/. It is more powerful then GAP, but it's not free.

GAP as a "fancy calculator" for Abstract Algebra.

What follows is a series of examples that show how your students can use GAP to check there work and do calculations. Of course one must stress to their students that they still need to know how to do the calculations by hand.

Example 1: Use Euler's Theorem to find the remainder of 7^{1001} when divided by 24.

Hopefully your students will work the problem correctly and get the answer 7.
The students can then check their work with GAP as follows:

```
gap> 7^1001 mod 24;
7
```

Example 2: Your students can use GAP to check their calculations when composing permutations. A very simple example of which is the following:

```
gap> (1,2,3)*(1,2);
(2,3)
```

Example 3: GAP can also be used to great effect in calculating the GCD of a number via the Euclidean Algorithm. For example to find the CGD of 4 and 15, i.e. (4,15), you type the following into GAP.

```
gap> Gcd(4,15);
1
```

A function your students will find even more interesting is Gcdex, applying this function to 4 and 15 we get

```
gap> Gcdex(4,15);
rec( gcd := 1, coeff1 := 4, coeff2 := -1, coeff3 := -15, coeff4 := 4 )
```

Here note that 4*4 - 1*15 = 1 (the Euclidean Algorithm!) and -15*4 + 4*15 = 0.

Using GAP as a tool for investigation in Abstract Algebra

GAP has many built in functions that can be used to investigate Abstract Algebra. The following are just a small sample of the functions which you and your students can use in your investigations.

Two of the most well know groups in Abstract Algebra are integers modulo n and the symmetric group on n letters. The following code defines *z10* to be the group of integers module 10 and *s4* to be the symmetric group of 4 letters.

```
gap> z10 := Integers mod 10;
(Integers mod 10)
```

```
gap> s4 := SymmetricGroup(4);
Sym( [ 1 .. 4 ] )
```

Now that we have to above groups defined we can use GAP to investigate them. To find the size of each group we use the command Size as follows:

```
gap> Size(z10);
10
gap> Size(s4);
24
```

GAP also has a nice function called IsAbelian which returns true if the group is abelian and false otherwise.

```
gap> IsAbelian(z10);
true
gap> IsAbelian(s4);
false
```

When your students are constructing multiplication tables for the symmetric group on 4 or 5 letters they can use GAP to quickly get all the elements in s4 using the command Elements.

```
gap> Elements(s4);
[ (), (3,4), (2,3), (2,3,4), (2,4,3), (2,4), (1,2), (1,2)(3,4), (1,2,3),
  (1,2,3,4), (1,2,4,3), (1,2,4), (1,3,2), (1,3,4,2), (1,3), (1,3,4),
  (1,3)(2,4), (1,3,2,4), (1,4,3,2), (1,4,2), (1,4,3), (1,4), (1,4,2,3),
  (1,4)(2,3) ]
```

The nice thing about this command is that it give them the elements, so they know what they're shooting for. But the command doesn't give them the multiplication table.

(Note: the first element () above isn't a type-o it's GAP's notation for the identity element.)

Your students can also use GAP to investigate the normal subgroups of a group. The command I find to be the most useful for this purpose is called NormalSubgroups. To find all the normal subgroups of s4 as defined above we type the following:

```
gap> NormalSubgroups(s4);
[ Sym( [ 1 .. 4 ] ), Group([ (2,4,3), (1,4)(2,3), (1,3)(2,4) ]),
  Group([ (1,4)(2,3), (1,3)(2,4) ]), Group(()) ]
```

In the above *Group(())* is the group containing only the identity, *Group([(1,4)(2,3), (1,3)(2,4)]* is the group generated by *(1,4)(2,3)* and *(1,3)(2,4)*, etc.

In the following example we construct the finitely presented group $\langle x, y | x^2 = y^2 = xyxy = e \rangle$ an investigate it's Size and determine if it is abelian.

```
gap> f:=FreeGroup(2);
<free group on the generators [ f1, f2 ]>
gap> g :=f/[f.1^2, f.2^2, f.1*f.2*f.1*f.2];
<fp group on the generators [ f1, f2 ]>
gap> Size(g);
4
gap> IsAbelian(g);
true
```

Finally to construct the Galois Field F_n in GAP one can use the command GaloisField or GF as follows.

```
gap> f2:= GaloisField(2^2);
GF(2^2)
gap> f4:= GF(2^4);
GF(2^4)
gap> Size(f2);
4
gap> Size(f4);
16
```

The above is just a small sample of the built in functions of GAP, all of which can be found at http://www.gap-system.org/. The website has extensive documentation and an online forum. You should have no trouble learning GAP and incorporating it into your Abstract Algebra class. If you have any questions, or want copies of my GAP labs please feel free to contact me at andersk@mwsc.edu.