Francesco Valenti

francesco.valenti1992@gmail.com

The Dihedral Group D_3 using GAP

A small tutorial on software approach to Group Theory

Grenoble, October 2015

Magistère de Physique, 2[eme] année - Théorie des Groupes

---

1. Introduction

This is a small tutorial on how to use the GAP software. We will use it study the Dihedral Group D_3, a group we've seen many times during our course with professor Schienbein. The GAP software can be downloaded for free; the code given in this tutorial can be simply copy-pasted into the GAP terminal.

In order to follow through this tutorial, you should input the code in your terminal as it is given, and check the output, since the resulting output will not be displayed in this document.

---

2. The GAP software and the D_3 group

D_3 is defined as the group generated by <b,c> with the composition law $c^3 = b^2 = bcbc = e$. This is said in 'human' language: in order to input the group into the software, we have to translate this statement into machine language.

GAP comes with a wealth of already defined groups and functions and we will rely on these. To input the D_3 group, we can use the following code:

```
f:=FreeGroup("b","c");   #the function FreeGroup lets us define some group f, with two generators
b:=f.1;  c:=f.2;                    #we call b and c the generators of f
rels:=[b^2,(b*c)^2,c^3];      #we define the composition law
D3:=f/rels;                   #we define D3
```

The sentences beginning with a # are comments, and don't affect the code.
D_3 is a very common group, and we can also find it in the vast GAP library just by typing

```
D3lib:=DihedralGroup(6);    #this defines D3lib as the dihedral group with 6 elements, which is D3
```

The GAP library gives us a powerful tool to check whether two groups are isomorphic. Groups in the library are precisely indexed, so if any group input by us or called from the library share the same library index, we can be sure that they're isomorphic. Let's check:

```
IdGroup(D3)=IdGroup(D3lib);    #we use the IdGroup function to check if these group are isomorphic
```

Notice that we used = instead of := ; we are not defining a group, but we're checking a boolean property. If up to this point everything was done correctly, the previous line should return "true". Therefore the groups are isomorphic, which means we successfully input the D_3 Group into the program.

The GAP software usually works best when dealing with permutations. The subgroup of the permutation group S_n which is isomorphic to D_3 is ((2,3),(1,2,3)), where (2,3) is the generator "b" and (1,2,3) is the generator "c".
GAP allows us to compute compositions of permutations, so let's check it:

```
(2,3)^2;
(1,2,3)^3;
((1,2)*(1,2,3))^2;
```

The following operations should all give (), the identity permutation, as a result, proving that this group is in fact isomorphic to D_3. Let's define it:

```
P3:=Group((2,3),(1,2,3));
```

And see if it is isomorphic to D_3:

```
IdGroup(D3)=IdGroup(P3);
```

The previous line should give 'true' as an answer. However, for any given group, GAP can tell us immediately the equivalent permutation (that is, the permutation isomorphic to said group):

```
P3:=IsomorphismPermGroup(D3);
```

When dealing with more intricated groups it's safer to work with the equivalent permutations; this is simply due to the software architecture of GAP, and the inevitable limitations of computing.

Now that we have defined D_3 (in fact, multiple times with equivalent definitions), we can study some of its properties.

```
Size(D3); #number of elements in D3
IsAbelian(D3);
IsSimple(D3);
```

A simple group is one having only itself and the trivial group as normal subgroups. Apparently D_3 is not simple, so we can look for normal subgroups. Let's try this:

```
NormalSubgroups(D3);
```

This doesn't work really well. As said before, sometimes it is more fruitful to work with equivalent permutations:

```
P3:=Group((2,3),(1,2,3));
```

```
NormalSubgroups(P3);
```

The center of a group is the set of the elements that commute with every other element in the group. One of these subgroups should be the center of D_3.. but which one?

```
Center(D3);
```

We see that D_3 has no nontrivial center. But D_4 has one, the group (e,c^2). One could conjecture that D_n doesn't have a nontrivial center if n is odd. Let's make a brute force approach:

```
D5:=DihedralGroup(10);
Center(D5);
D7:=DihedralGroup(14);
Center(D7);
D11:=DihedralGroup(22);
Center(D11);
D31:=DihedralGroup(62);
Center(D31);
```

This really looks like it's the case! A formal proof can't be done on a computer, but this gives us a tool to come up with more sound conjectures, checking a handful of cases in almost no time. A formal proof is given at the end of this document.

Anyway, let's go back to D_3.

We can find its conjugacy classes:

```
ConjugacyClasses(P3);    #again, we work on the equivalent permutation in order to get better results
```

There is an even niftier way to do so:

```
L:=LatticeSubgroups(P3);
ConjugacyClassesSubgroups(L);
```

We can obtain its Cayley table:

```
Display(MultiplicationTable(P3));
```

And even its character table:

```
Display(CharacterTable(P3));
tbl:=CharacterTable(P3);
SizesConjugacyClasses(tbl);   #this tells us the multiplicity of the conjugacy classes
```

This is only a glimpse of what the GAP software is capable of, and yet it already provided us an almost immediate tool to study properties that took us quite some time when done by hand.

3. Proof that the dihedral group D_n doesn't have a nontrivial center if n is odd

(Courtesy of Karl Pelka)

$cb = bc^k c^{n-k-1} = bc^{n-1}$

$bc^k = bc^{k+1} = b = b(bcbc)c^k = cbc^{k+1}$

$c^p b = bc^{p(n-1)}$

Suppose that p(n-1) mod n = p(n-1)-nz

p mod n = p-nx

p-nx = p(n-1)-nz

p = (n/2)(p-(z-x)) where p,x,z belong to Z, thus n odd → p doesn't belong to Z

Suppose that $(bc^p)b = b(bc^p)=c^p$

$bc^p b = c^{p(n-1)}$

p mod n = p(n-1)mod n

p = (n/2)(p-(z-x)), thus n odd → p doesn't belong to Z, which proves our theorem.