

### 3 Chapter: Finite Groups; Subgroups

In order to do the exercises in this section you will need to read into GAP the files “ulist” and “cyclic.” The file “cyclic” was written by Loren Larson at St. Olaf College and then revised per comments from Alexander Hulpke. The file “ulist” should already be in your gap4r4 folder. Fetch the file “cyclic” from the web site for this manual and place the file in the gap4r4 folder. The appendix at the end of this chapter contains a print out of the file “cyclic.”

GAP has many useful features that allow you to examine subgroups:

```
gap> G:= DihedralGroup(IsPermGroup, 16);
Group([ (1,2,3,4,5,6,7,8), (2,8)(3,7)(4,6) ])
gap> Elements(G);
[ (), (2,8)(3,7)(4,6), (1,2)(3,8)(4,7)(5,6), (1,2,3,4,5,6,7,8), (1,3)(4,8)(5,7),
(1,3,5,7)(2,4,6,8), (1,4)(2,3)(5,8)(6,7), (1,4,7,2,5,8,3,6), (1,5)(2,4)(6,8),
(1,5)(2,6)(3,7)(4,8), (1,6)(2,5)(3,4)(7,8), (1,6,3,8,5,2,7,4), (1,7)(2,6)(3,5),
(1,7,5,3)(2,8,6,4), (1,8,7,6,5,4,3,2), (1,8)(2,7)(3,6)(4,5) ]
gap> a:= G.1;
(1,2,3,4,5,6,7,8)
gap> b:= G.2;
(2,8)(3,7)(4,6)
gap> H:= Subgroup(G, [a]);
Group([ (1,2,3,4,5,6,7,8) ])
gap> Elements(H);
[(), (1,2,3,4,5,6,7,8), (1,3,5,7)(2,4,6,8), (1,4,7,2,5,8,3,6),
(1,5)(2,6)(3,7)(4,8), (1,6,3,8,5,2,7,4), (1,7,5,3)(2,8,6,4),
(1,8,7,6,5,4,3,2)]
```

The first command above assigns the name  $G$  to the dihedral group of order 16 (the group of symmetries of a regular 8-gon). From the next command listing the elements in  $G$ , we can see that both  $(1, 2, 3, 4, 5, 6, 7, 8)$  and  $(1, 8)(2, 7)(3, 6)(4, 5)$  are elements of  $G$ . More generally,  $(1, 2, 3, \dots, n)$  represents a rotation of  $360/n$  degrees; for  $n$  even,  $(1, n)(2, n-1)(3, n-2)\dots(\frac{n}{2}, \frac{n}{2}+1)$  is a reflection; and for  $n$  odd,  $(1, n)(2, n-1)\dots(\frac{n-1}{2}, \frac{n-1}{2}+2)$  is a reflection. The next two commands assign the names  $a$  and  $b$  to two generators of  $G$ . The next command assigns the name  $H$  to the cyclic subgroup of  $G$  generated by  $a$ . Notice that  $H$  is a subgroup of  $G$  of order 8.

```
gap> K:= Subgroup(G, [a,b]);
Group([ (1,2,3,4,5,6,7,8), (2,8)(3,7)(4,6) ])
gap> Size(K);
16
```

This first command above assigns the name  $K$  to the subgroup of  $G$  generated by  $a$  and  $b$ . That is,  $K$  is the subgroup of  $G$  obtained by taking all possible finite strings of  $a$ 's and  $b$ 's. Notice that after this command GAP echoes the generators of the subgroup. By typing in  $\text{Elements}(K)$  or  $\text{Size}(K)$  we can see that  $K = G$ .

In the line `gap> K:= Subgroup(G, [a,b]);` the use of `[a,b]` is a list in GAP. In general, square brackets enclose lists. In this case, we are listing the generators of  $K$ .

```
gap> c:= (1,5)(2,6)(3,7)(4,8);;
gap> L:= Subgroup(G, [a,c]);
Group([ (1,2,3,4,5,6,7,8), (1,5)(2,6)(3,7)(4,8) ])
gap> Size(L);
8
```

Notice that the subgroup  $L$  is a proper subgroup of  $G$ . In fact,  $c$  is a power of  $a$ , as we can see from the list of elements of  $H$ . Thus  $L$  is a subgroup of  $H$ . But  $L$  and  $H$  have the same order, so  $L = H$ . Also notice the use of a double semicolon at the end of the line defining  $c$ . This causes GAP not to echo the definition of  $c$  on the next line. (Compare the lines defining  $a$  and  $b$  with the line defining  $c$ .)

```
gap> M:= Subgroup(G, [a^2,b]);
Group([ (1,3,5,7)(2,4,6,8), (2,8)(3,7)(4,6) ])
gap> Elements(M);
[ (), (2,8)(3,7)(4,6), (1,3)(4,8)(5,7), (1,3,5,7)(2,4,6,8), (1,5)(2,4)(6,8),
(1,5)(2,6)(3,7)(4,8), (1,7)(2,6)(3,5), (1,7,5,3)(2,8,6,4) ]
```

Notice  $M$  is another subgroup of  $G$  of order 8, not equal to  $H$ :

```
gap> M=H;
false
```

The function `cyclic(n,a)` which is in the file “cyclic” produces the list of elements in the cyclic subgroup of  $U(n)$  generated by the element  $a$  in  $U(n)$ . For example,

```
gap> cyclic(15,7);
[ ZmodnZObj( 7, 15 ), ZmodnZObj( 4, 15 ), ZmodnZObj( 13, 15 ),
ZmodnZObj( 1, 15 ) ]
```

gives the subgroup of  $U(15)$  generated by 7. (The output `ZmodnZObj( 7, 15 )`, for example, means the element  $7 \bmod 15$  in  $U(15)$ .) If you use this function incorrectly and try to generate a subgroup generated by  $a$  when  $a$  is not in  $U(n)$ , this function will return empty brackets:

```
gap> cyclic(15,3);
[ ]
```

The following is a list of some other commands that you might find useful.

- 1) The command to find the center of the group  $G$  is `Center(G)`.

- 2) The command to find the centralizer of an element  $a$  in a group  $G$  is `Centralizer(G,a)`.
- 3) The command to find the order of an element  $a$  in a group  $G$  is `Order(a)`.
- 4) The command `IsAbelian(G)` tells you whether or not the group  $G$  is Abelian.
- 5) The command `IsCyclic(G)` tells you whether or not the group  $G$  is cyclic.

There is no need for you to memorize a large collection of GAP commands. Just type in ? followed by a key word describing what you want GAP to do, and the software will provide helpful comments and examples on using commands. For example, say we want to find the order of an element in a group and wonder exactly how to type this in GAP. Type:

```
gap> ?order
```

GAP then provides many possible help topics:

```
Help:  several entries match this topic - type ?2 to get match [2]
[1] Tutorial:  Order
[2] Reference:  Order
[3] Gpd (not loaded):  Order
[4] kbmag (not loaded):  Order
[5] XMod (not loaded):  Order
[6] Reference:  order!  of a group
[7] Reference:  Order!of a class function
[8] Reference:  order!of the prime residue group
[9] Reference:  order!of a list, collection or domain
[10] Reference:  Orderings on families of associative words
[11] Reference:  Orderings
[12] Reference:  OrderMod
[13] Reference:  OrderedPartitions
[14] Reference:  ordering!booleans
[15] Reference:  ordering!of records
[16] Reference:  OrderingsFamily
[17] Reference:  OrderingByLessThanFunctionNC
[18] Reference:  OrderingByLessThanOrEqualFunctionNC
[19] Reference:  OrderingOnGenerators
[20] Reference:  OrderOfRewritingSystem
[21] Reference:  OrderingOfRewritingSystem
[22] Reference:  OrdersTom
[23] Reference:  OrdersClassRepresentatives
[24] Extending:  ordered partitions!internal representation
[25] GRAPE (not loaded):  OrderGraph
[26] GRAPE (not loaded):  OrderGraph
[27] GUAVA (not loaded):  order of polynomial
```

```

[28] kbmag (not loaded): OrderingOfKBMAGRewritingSystem
[29] kbmag (not loaded): OrderingOfRewritingSystem
[30] MONOID (not loaded): OrderPreservingSemigroup
[31] RCWA (not loaded): Order of an rcwa permutation
[32] RDS (not loaded): Ordered signatures by quotient images
[33] RDS (not loaded): Ordered signatures using representations
[34] RDS (not loaded): Ordered Signatures
[35] RDS (not loaded): OrderedSigsFromQuotientImages
[36] RDS (not loaded): OrderedSigInvariant
[37] RDS (not loaded): OrderedSigs
[38] RDS (not loaded): OrderedSignatureOfSet

```

It looks like the sixth one is the one we want so type:

```
gap> ?6
```

GAP then provides a description of the `Order` command and an example.

### *Exercises*

Use GAP to help you work the following exercises.

3.1 Determine whether the group  $U(n)$  is cyclic for  $n = 3, 9, 27, 5, 25, 125, 7, 49, 343$ . (Use the function `cyclic` discussed above but not the command `IsCyclic`.) Make a conjecture. Test your conjecture for other values of  $n$ .

3.2 Determine whether the group  $U(n)$  is cyclic for  $n = 6, 18, 54, 10, 50, 250, 14, 98, 686$ . Make a conjecture.

3.3 Determine whether the group  $U(n)$  is cyclic for  $n = 8, 16, 32$ . Modify your conjectures above if necessary. Test your conjecture for other values of  $n$ .

3.4 Determine whether the group  $U(n)$  is cyclic for  $n = 12, 20, 28, 44, 52, 15, 21, 33, 39, 51, 57, 69, 35, 55, 65, 85$ . Modify your conjectures above if necessary.

3.5 Must the centralizer of an element of a group be Abelian? If not, give an example in  $D_n$  for some  $n$ .

3.6 Must the center of a group be Abelian? If not, give an example in  $D_n$  for some  $n$ .

Recall the file “ulist” contains a function that lists all the elements in the group  $U(n)$ . Read this file into GAP. In the following we are going to investigate the relationship between the order of an element and the order of the inverse of that element. Consider  $U(15)$ , which is a **subset** of  $\mathbf{Z}_{15}$ .

```

gap> e := Elements(ulist(15));
[ ZmodnZObj( 1, 15 ), ZmodnZObj( 2, 15 ), ZmodnZObj( 4, 15 ), ZmodnZObj( 7, 15 ),

```

```
ZmodnZObj( 8, 15 ), ZmodnZObj( 11, 15), ZmodnZObj( 13, 15), ZmodnZObj( 14, 15) ]
gap> e[3];
ZmodnZObj( 4, 15 )
```

From the above output we see that  $U(15)$  contains the numbers 1, 2, 4, 7, 8, 11, 13, and 14 (mod 15). The first command above assigns the name `e` to the list of elements in  $U(15)$ . Since 4 (mod 15) is the 3rd element in this list we can then refer to 4 as `e[3]`, as is done in last above GAP command. We can now determine the order of 4 in  $U(15)$ .

```
gap> Order(e[3]);
2
gap> Order(Inverse(e[3]));
2
```

### Exercises

3.7 Compute the orders of the elements 3, 7, 53, and 61 in  $U(100)$ . Compute the orders of the inverses of these elements.

3.8 Compute the orders of the elements 3, 13, 153, and 317 in  $U(430)$ . Compute the orders of the inverses of these elements.

3.9 Pick a symmetric group  $S_n$  for some particular  $n$  and call it  $G$  in GAP. (The command to create the symmetric group  $S_5$ , for example, in GAP is `SymmetricGroup(5)`.) The command `Random(G)` will give you a random element in  $G$ . Find the order of a random element in  $G$  and the order of its inverse. Repeat this exercise for at least two other random elements of  $G$  and at least two other symmetric groups.

3.10 Make a conjecture about the relationship between the order of an element and the order of the inverse of that element.

3.11 Pick a symmetric group  $S_n$  for some particular  $n$ . Find the orders of two random elements in your group and the order of their product. Repeat this exercise for at least four other pairs of random elements of  $S_n$  and at least two other symmetric groups. Based on your results, what do you think we can say about the order of  $ab$  in terms of the orders of  $a, b \in S_n$ .

3.12 Repeat Exercise 3.11 for the groups  $GL(2, \mathbf{Z}_n)$  where  $n$  is some particular prime. (The command `GL(2,p)` in GAP sets up the group  $GL(2, \mathbf{Z}_p)$ .)

The remainder of this chapter is not needed in the sequel. It is intended for students who would like to learn more about GAP.

If you would like to see how a predefined function GAP is being computed you can do the following. For example, suppose we want to see how GAP is executing the `IsCyclic` function. Type:

```
gap> G:= DihedralGroup(IsPermGroup, 16);;
```

```

gap> obj:=[G];;
gap> code:= ApplicableMethod(IsCyclic,obj,1);
#I Searching Method for IsCyclic with 1 arguments:
#I Total: 6 entries
#I Method 3: 'IsCyclic', value: 22
function( G ) ... end

```

You can then have the code for IsCyclic printed on the screen:

```

gap> Print(code);
function ( G )
  if Length( GeneratorsOfGroup( G ) ) = 1 then
    return true;
  else
    return TRY_NEXT_METHOD;
  fi;
  return;
endgap>

```

Some predefined functions require more than one argument. All arguments need to be included in the third entered line above (the line starting with `gap> code:=`).

### Appendix for Chapter 3

The following is the file “cyclic” which is used in this chapter.

```

cyclic:= function(n,a)
local x, b, o ;
x:= [];
b:= 1;
o:= One(Integers mod n);
  if Gcd(n,a) = 1 then
    repeat
      b:= b*a mod (n);
      Add(x,b);
    until b=1;
  fi;
return x*o;
end;

```