Math 103A – Winter,2004
Professor John J Wavrik

Permutations and Isomorphisms

A permutation of {1, …, n } is a 1-1, onto mapping of the set to itself. Most books initially use a bulky notation to describe a permutation: The numbers 1..n are put on one row and the images of these elements under the permutation are put below. Thus

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$ describes the permutation which sends $1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 3$.

This is quickly abandoned in favor of a 1 line "cycle notation" where the same permutation would be denoted (1 2). Groups32 uses cycle notation.

It may be of interest for you to know that Groups32 internally stores permutations in the bulky notation and converts them, for input and output, to the cycle notation.[1]

---

[1]   If anyone feels that it would be useful to allow input and output in the 2-line form, please let me know and I will put that feature in.

To use permutations, you must enter a sub-package of Groups32 called PERMGRPS.[2] Notice that both the prompt and the list of commands changes:[3]

CREATE   ELEMENTS   HELP   INFO
INSTALL   MAIN   MULTIPLY   QUIT
X

PERM >>

You can find out what each command as usual: type X (or INFO) and the command. This will give you a description.

---

[2]   The rationale for doing this as a subpackage is to reduce clutter in the main package.

[3]   The Telnet version has two additional commands: `Left->Right` and `Right->Left` to change the direction of multiplication. The lab version is preset for whatever is used in the textbook for the course.

1. Multiplying Permutations

IN THESE EXAMPLES WE MULTIPLY RIGHT TO LEFT

```
PERM>> MULTIPLY
Subgroup of Sn -- what is n?  Number 3
     Put in a product of cycles.
     End with a blank line
Cycles (1 2 3)(1 2 3) = (1 3 2 )
Cycles (1 2)(1 3) = (1 3 2 )
Cycles (1 3)(1 2) = (1 2 3 )
Cycles
```

This command will keep multiplying cycles until you put in a blank line. The input is a product of cycles. The output is a product of disjoint cycles.

2. *Subgroup of S_n generated by given permutations*

The subgroup generated by a set of permutations is the smallest subgroup of S_n which contains them. This subgroup can be found by the CREATE command[4]. Each of the generating permutations is entered on a separate line. The input is terminated by a blank line.

```
PERM>> CREATE
Subgroup of Sn -- what is n?   Number 5
     Put in generators as product of cycles.
     End with a blank line
Generator (1 2)(3 4)
Generator (1 3)
Generator
Group is of order 8
A  ()                B  (2 4 )              C  (1 2 )(3 4 )
D  (1 2 3 4 )        E  (1 3 )              F  (1 3 )(2 4 )
G  (1 4 3 2 )        H  (1 4 )(2 3 )
```

---

[4] It is easy to generate groups too big for manipulation by Groups32. You will receive a notice if your group is too big.

The subgroup generated by { (1 2)(3 4), (1 3) } is of order 8 and its elements are listed.
If you wish to use the features of the main package to investigate the group you generated, you must INSTALL it to take the place of one of the groups 1-5. Then you must return to the main package by typing MAIN:

```
PERM>> INSTALL
     Install as table k (1..5)  Number 1
PERM>> MAIN

  CENTER        CENTRALIZER   CHART         CONJ-CLS
  COSETS        EVALUATE      EXAMPLES      GENERATE
  GROUP         HELP          INFO          ISOMORPHISM
  LEFT          NORMALIZER    ORDERS        PERMGRPS
  POWERS        QUIT          RESULT        RIGHT
  SEARCH        STOP          SUBGROUPS     TABLE
  X

  G1>>
```

Here I have chosen to have this group installed as group1. When I return to MAIN, the list of commands is printed and the prompt changes to show the current group.

```
G1>> ORDERS    for Group Number 1

Group number 1 of Order 8
     1 elements of order  1:   A
     5 elements of order  2:   B C E F H
     2 elements of order  4:   D G
     0 elements of order  8:

G1>> EVALUATE   (use ' for inverse) bd= H
G1>> EVALUATE   (use ' for inverse) db= C
G1>> POWERS    for element D
A D F G
```

Compare with the letters assigned to permutations in this group. You will see that the results just obtained are consistent:

```
A  ()                B  (2 4 )           C  (1 2 )(3 4 )
D  (1 2 3 4 )        E  (1 3 )           F  (1 3 )(2 4 )
G  (1 4 3 2 )        H  (1 4 )(2 3 )
```

Notice, for example, that D = (12 3 4) is of order 4 and its powers are D^1=D, D^2=F, D^3=G, D^4=A

An Algorithm

   We know that the subgroup generated by a set of elements is obtained by evaluating all words in the elements and their inverses. There are an infinite number of words. If the group is finite, there can only be a finite number of elements. Thus we must reach a point where taking further words does not produce more elements.

Suppose, for example, we are trying to find the subgroup of S_n generated by permutations a, b[5]. Call the subgroup H. It must contain the identity and a and b. It must also contain all products that can be formed using the letters a and b. The algorithm starts with the set $S = \{a,b\}$. We multiply the elements of S on the right by a and by b (so we get words of two letters: aa, ba, ab, bb. Evaluate and add any new elements to S. Multiply the new elements by a and b and add any newly obtained elements to S (we are now checking words of length 3). Repeat this process. When no new elements are obtained we have H = S.

---

[5]   We will not use inverses in forming words in this case. Since the group is finite, the inverse of any element is a power of the element. So inverses will automatically appear.

Isomorphism

The ISOMORPHISM command in Groups32 allows you to find an explicit isomorphism between two groups. We have just installed a permutation group as group 1:

```
A  ()                B  (2 4 )           C  (1 2 )(3 4 )
D  (1 2 3 4 )        E  (1 3 )           F  (1 3 )(2 4 )
G  (1 4 3 2 )        H  (1 4 )(2 3 )
```

This is the subgroup of S_4 generated by { (1 2)(3 4), (1 3) }. It is of order 8 and its elements are listed. Once it has been installed as group 1 we can use the commands of Groups32 to find out about it.

This should be isomorphic to one of the groups of order 8 in the collection of groups that comes with Groups32. Let's try to find which one.

```
G1>> CHART   Order of Groups (1-32 or 0) Number 8
       10   11   12   13*  14*
         There are 5 Groups of order 8
         3 abelian and 2 non-abelian
```

There are 5 possible choices. What do we need to know about our Group 1 to decide? One possibility is to find out if Group 1 is abelian or not. We can use the CENTER command:

```
G1>> CENTER    of Group Number 1
{ A F }
```

This tells us that the center of G1 has only two elements. Thus G1 is not abelian. We might then check which of the non-abelian groups (13 or 14) has the same size center.

```
G1>> CENTER    of Group Number 13
{ A C }
G13>> CENTER    of Group Number 14
{ A C }
```

It appears that both of these groups have a center of order 2 – so this will not help us decide.
We know that under isomorphism the orders of elements are preserved. Let us look at the orders of the elements for Group 1 and also for Groups 13 and 14:

```
G14>> ORDERS    for Group Number 1                    G1>> ORDERS    for Group Number 13

Group number 1 of Order 8                             Group number 13 of Order 8
    1 elements of order  1:    A                           1 elements of order  1:    A
    5 elements of order  2:    B C E F H                    5 elements of order  2:    C E F G H
    2 elements of order  4:    D G                          2 elements of order  4:    B D
    0 elements of order  8:                                 0 elements of order  8:


                                                      G13>> ORDERS    for Group Number 14

                                                      Group number 14 of Order 8
                                                          1 elements of order  1:    A
                                                          1 elements of order  2:    C
                                                          6 elements of order  4:    B D E F G H
                                                          0 elements of order  8:
```

Notice that the distribution of orders for Group 1 is the same as for Group 13 but not for Group 14. This suggests that we should try constructing an isomorphism between Group 1 and Group 13. It also even suggests candidates for a mapping (notice that D in G1 would have to be sent to B or D in G13, for example).

The ISOMORPHISM command asks for the numbers of the source and target groups (we put in 1 and 13). We are then shown a succession of frames asking which letter in the source group is to be mapped to which in the target.

```
G14>> ISOMORPHISM   from Group Number 1 to Group Number 13
```

```
         Top: Group # 1              BOTTOM: Group # 13

         A B C D E F G H

         Send:  D           To:  D
```

We have sent D to D. The ISOMORPHISM command is smart enough to deduce consequences (D*D in G1 must get sent to D*D in G13, A must get sent to A, etc). So we now have the entire subgroup generated by D in G1 mapped to the subgroup generated by D in G13

```
Top: Group # 1              BOTTOM: Group # 13

A B C D E F G H
A     D   C B

Redo the last assignment (y/n/q)?
```

Please notice that you are given a chance to change your mind. You can redo the last assignment or quit working on the isomorphism. We will keep this assignment (so say "n")

We now try to map B (an element of order 2) to one of the elements of order 2 in G13. You might experiment to see if it makes a difference. We will sent B to E.
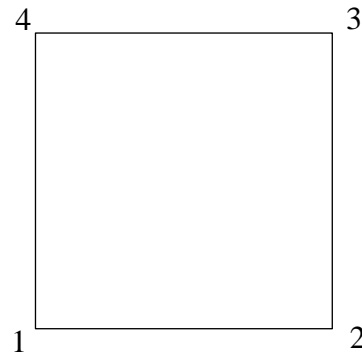
```
G14>> RESULT
        Top: Group # 1              BOTTOM: Group # 13

        A B C D E F G H
        A E F D G C B H
```

Here is the explicit isomorphism.

[There is a difference in some versions of Groups32. The "stand alone" version pops up a separate window for the isomorphism. This allows you to see information on your original Groups32 sheet. When the isomorphism is found the window disappears and you type the RESULT command to see the isomorphism. The Internet version handles the isomorphism in the Groups32 main window (but erases the window). You will see the isomorphism without typing RESULT.]

Appendix

There are some other matters that are of interest about our permutation group. It is the dihedral group D_4.
This is the group of symmetries of a square: the permutations of the vertices which send the square to itself. It is
interesting to spend some time connecting the algebra of the group with the geometry of the square.

```
4                    3




1                    2
```

```
A   ()                    B   (2 4 )                C   (1 2 )(3 4 )
D   (1 2 3 4 )            E   (1 3 )                 F   (1 3 )(2 4 )
G   (1 4 3 2 )            H   (1 4 )(2 3 )
```

The generator C = (1 2)(3 4) is a reflection in the vertical axis. We took E = (1 3) as the other generator which is a reflection along the diagonal joining vertices 2 and 4. Notice that D = (1 2 3 4) is a rotation. It is reasonable to wonder how a rotation can be obtained as a combination of reflections (use the MULTIPLY command to find out).

We may also wonder about the subgroups of this group, and their geometric significance.

```
G1>> SUBGROUPS    of Group Number 1
   * = Normal subgroup
   Generators              Subgroup
   0  { }                  *{ A }
   1  { B }                 { A B }
   2  { C }                 { A C }
   3  { E }                 { A E }
   4  { F }                *{ A F }
   5  { H }                 { A H }
   6  { B E }              *{ A B E F }
   7  { D }                *{ A D F G }
   8  { C F }              *{ A C F H }
   9  { B C }              *{ A B C D E F G H }
```

We found, for example, that {A F} is the center of this group. F may look like a flip of some sort – but it is actually a rotation of the square by 180 degrees. Why should this commute with everything?

Notice that this chart shows that the whole group is generated by B and C. What are other generators? (notice that the GENERATE command returns the subgroup generated by a given set of generators).

Why are the subgroups marked * normal and the others not?