Math 103A – Winter,2001
Professor John J Wavrik

Permutations

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 $1 \to 2$, $2 \to 1$, $3 \to 3$.

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]

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.

---

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.

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 )
```

The subgroup generated by { (1 2)(3 4), (1 3) } is of order 8 and its elements are listed.

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

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.