16 Algebra

16.5 Permutation Groups (7 units)

This project is self-contained, building on theory covered in the Part IA course Groups. *Some knowledge of the groups part of the Part IB course* Groups, Rings and Modules *would be useful.*

1 Introduction

We are given a set of permutations of $X = \{1, \ldots, n\}$. They generate a finite permutation group $G \leqslant S_n$. The aim of this project is to replace the set of generators of G with another generating set of a very special nature. This allows us to deal with various questions. You may assume for programming purposes that $n \leqslant 20$.

2 Permutations

A permutation π of X is a bijective function from X to X. If x is an element of X then the image of x under π is written πx. If π_1 and π_2 are permutations then their product $\pi_1 \cdot \pi_2$ maps x to $\pi_1(\pi_2 x)$. The set of all permutations of the set $X = \{1, \ldots, n\}$ is the symmetric group S_n. If π is a permutation and $y = \pi y$ then y is called a *fixed point of* π.

> **Question 1** Write procedures to compute the inverse π^{-1} of a permutation π and the product $\pi_1\pi_2$ of two permutations π_1 and π_2. What is the complexity of your method for computing inverses (as a function of n)?

3 Groups

Suppose the permutation group G is generated by permutations $\pi_1, \ldots, \pi_k$. First we reduce the number of generators with the Stripping Algorithm of Sims. Let A be an $n \times n$ array of permutations which is initially empty.

Suppose we have already put the first $l-1$ permutations into the array. If π_l does not fix 1 and the $\pi_l(1)$th entry in the first row is still empty then put π_l there. Suppose the $\pi_l(1)$th entry is the permutation g. Then modify π_l to be $g^{-1}\pi_l$ so the new π_l fixes 1. Go to the second row.

If π_l does not fix 2 and the $\pi_l(2)$th entry in the second row is still empty then put π_l there. If the entry is g then modify π_l to be $g^{-1}\pi_l$ which hence fixes 1 and 2. Go on to the third row ...

If we reach the last row then we must have produced the trivial permutation which can be omitted from the generating set.

Once a permutation is placed in the array, or deemed to be the trivial permutation, we go on to try to place the next permutation in the array.

> **Question 2** Show that the modified set of permutations generates the group G. Give an upper bound for the size of the modified set of generators and for the number of operations needed to complete the algorithm. (As a function of n and the size of the original generating set, noting that, e.g., storing a permutation is $O(n)$ operations.)

Question 3 Write a procedure which computes the array of a permutation group given by a set of generators. It should receive a set of permutations as input and give a set of permutations as output, which generate the same group and are in the above reduced form. (Here, as elsewhere in this project, you should give some examples to demonstrate that your program is working correctly.)

4 Orbit and Stabilizer

Let G be a permutation group of X. If $\alpha \in X$ then the set $A = \{\beta \in X \mid \exists g \in G, g\alpha = \beta\}$ is called the *orbit* of α (under G). If $\beta \in X$ is in the orbit of α then an element $g \in G$ is called a *witness* of this if $g\alpha = \beta$. It is easy to see that β is in the orbit of α if and only if the orbit of β is the same as the orbit of α. Hence different orbits are disjoint and the orbits form a partition of X.

The *stabilizer* of an element α in X is $G_\alpha = \{g \in G \mid g\alpha = \alpha\}$. It is a subgroup of G.

Question 4 Write down a bijection between the set of left cosets of G_α in G and the orbit of α. State the orbit-stabilizer theorem.

Question 5 Write a procedure which computes the orbit with witnesses of a given element under a permutation group G generated by a given set of permutations. It should receive as input a set of permutations and an element $\alpha \in X$ and should return as a output a list of elements forming the orbit of α, together with a witness in each case. Briefly explain how your procedure works.

5 Schreier's Theorem and the final algorithm

Suppose G is a permutation group of X, given with a set of generators Y, α is an element of X and T is a complete set of left coset representatives of G_α in G. Let the surjective map $\varphi\colon G \to T$ be defined via $g\alpha = \varphi(g)\alpha$.

Question 6 Let x be an element of G_α. Write $x = y_r \dots y_1$ with each y_i an element of Y. Let t_1 be the element of T belonging to G_α. Let $t_{i+1} = \varphi(y_i t_i)$ for $i = 1, 2, \dots, r$. Show that $t_{r+1} = t_1$. Deduce that G_α is generated by the set of elements:

$$\{\varphi(yt)^{-1} \cdot y \cdot t \mid y \in Y, t \in T\}.$$

This is a special case of Schreier's Theorem.

Question 7 Write a procedure which computes a generating set of a stabilizer of a permutation group given with a set of generators. It should receive a set of permutations and an element α as input and give a set of permutations as output which generate the stabilizer. Use Question 5 to obtain T, then use Schreier's Theorem and finally reduce the set of generators with the Stripping Algorithm. Comment on the complexity of your algorithm.

Question 8 Write a program which computes the order of a permutation group G given with a set of generating permutations. The program should receive a set of permutations as input and give a natural number as output which is the order of G. You should first reduce the number of generators with the Stripping Algorithm, and recursively find a nontrivial orbit and use the previous question until you reach a subgroup of order 1. Use the orbit-stabilizer theorem in the recursive part to find the order of G.

Give an estimate for the number of operations required by this algorithm. Use this program to compute the order of the groups generated by the following sets:

- $\{(1,2),(2,3),(3,4),(4,5),(5,6),(6,7)\}$
- $\{(1,4,7),(2,5,8),(3,6,9,1)\}$
- $\{(1,2,3,4,5),(5,6,7,8,9,10,11)\}$
- $\{(1,11),(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)\}$
- $\{(3,9,15),(1,7),(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)\}$

You should make a note of the group order and number of generators (before and after stripping) for each of the subgroups computed. Briefly discuss how the algorithm might perform if we did not use the Stripping Algorithm at each stage.

Are any of the generators for the groups listed above redundant?

