Orbiter Programmer's Guide

Anton Betten

February 17, 2019

Abstract

We discuss how to use the program system Orbiter for the classification of combinatorial objects.

1 Working with groups

An abstract group can have many different permutation actions. In Orbiter, every group comes with at least one particular permutation action. Hence, we adopt the convention that if we say "group" we mean "permutation group."

The class `action` is used to represent a group with a specific permutation action. It is possible to have multiple permutation actions for the same group: one instance of class `action` is used for each action. The group has a base and strong generating set. A group element can be stored in different ways:

(a) as an array of consecutive integers (`int`);

(b) as an array of consecutive characters (`chars`);

(c) as a numerical value using indexing of the elements of a group;

The second form is called the compact form. The field `action::elt_size_in_int` stores how many ints are needed to store one group element. The field `action::coded_elt_size_in_char` stores how many chars are needed to store one group element in the compact form. The field `action::degree` stores the permutation degree of the action. The class `vector_ge` (vector of group elements) can be used to store an array of group elements. The class `strong_generators` can be used to store a strong generating set of a group in a given action. The basis is stored as part of the `action` class.

2 Computing orbits

In Orbiter, there are many different ways to compute orbits of groups. One important technique is the use of Schreier trees. The class `schreier` can be used to compute the

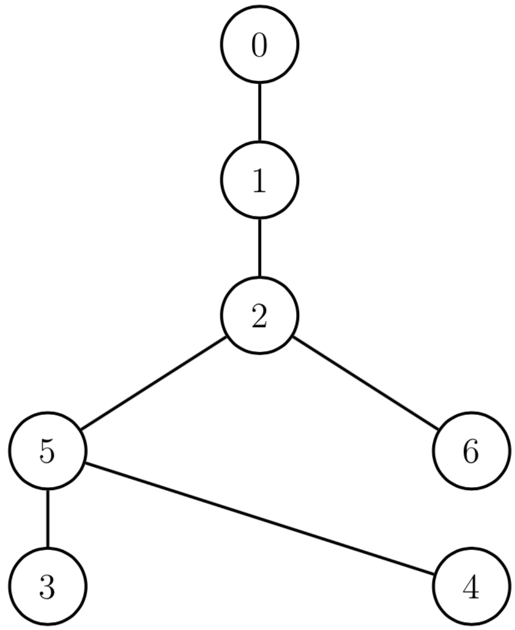

Figure 1: The Schreier tree for PGL$(3, 2)$ acting on PG$(2, 2)$.

Schreier forest of a particular group action. The storage requirement of a Schreier tree is proportional to the degree of the action. This means that Schreier trees should only be used for "small" degree actions. For larger actions, poset classification techniques are often better. However, poset classification only works if there is a poset structure on the underlying set on which we act. This may or may not be the case. It depends very much on the nature of the objects which we classify what kind of tools should be used.

Let us consider an example. We consider the action of PGL$(3, 2)$ on the seven points of PG$(2, 2)$. The schreier data structure is displayed in Table 1. The graphical representation of the Schreier tree is shown in Figure 1.

4 generators in action $PGL(3,2)$ of degree 7:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

1 orbits:
orbits under a group with 4 generators acting on a set of size 7:
Orbit 0 / 1 : { 0, 1, 2, 5, 6, 3, 4 } of length 7

i	*orbit*	*orbitinv*	*prev*	*label*	*cosetrep*
0	0	0	-1	-1	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
1	1	1	0	3	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
2	2	2	1	2	$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
3	5	5	2	0	$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
4	6	6	2	1	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
5	3	3	5	1	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
6	4	4	5	2	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 1: The Schreier structure for PGL$(3,2)$ acting on PG$(2,2)$.