AIMS DMG Sage Demonstration 09
AIMS 2013-14: Designs, Matroids and Graphs

Rob Beezer
University of Puget Sound

Nancy Neudauer
Pacific University

January 9, 2014

1 Automorphism Groups in Sage

We can build permutation groups from scratch in Sage, but it is more interesting to get them as structure-preserving bijections (permutations) of combinatorial structures. That is worth saying again: "structure-preserving bijections of combinatorial structures".

Build a small graph and compute its automorphism group. *Note:* the vertices of Sage graphs by default are integers starting at zero, which is very convenient for programming. The automorphism group of such a graph properly permutes these names for the vertices. However, the tradition for generic permutation groups is to permute integers, starting from one. This is not a problem — just be aware of it when you see it.

```
C4 = graphs.CycleGraph(4)
C4.plot()
```

```
A = C4.automorphism_group()
A
```

```
A.list()
```

Another graph, a bit more interesting.

```
cube = graphs.CubeGraph(3)
cube.plot()
```

```
cube.plot(layout="spring")
```

```
B = cube.automorphism_group()
B
```

On the one hand it is nice to see the permutations as functions on the names of the vertices (binary strings of length 3). On the other hand it is a bit of a mess and a bit of a distraction. We make a copy of the graph, then relabel its vertices with integers and go from there.

```
cube_relabeled = copy(cube)
cube_relabeled.relabel()
cube_relabeled.automorphism_group()
```

We can use either group for computations, the labeling will not matter for many items of interest. I have `B` saved already.

```
B.order()
```

```
B.list()
```

Now lets turn our attention to automorphisms of designs.

```
had = designs.HadamardDesign(15)
had
```

```
had.is_block_design()
```

There are 15! possible permutations of the points of the design. How big a number is that?

```
factorial(15)
```

```
factorial(15).log(10)
```

```
N(factorial(15).log(10))
```

```
C = had.automorphism_group()
C
```

```
C.order()
```

```
factor(C.order())
```

The matroid routines in Sage are very new. Automorphism groups are not implemented yet (Version 6.0), but that addition is planned. Perhaps it will build off of Sage's excellent automorphism routines for graphs (as much of the other automorphism routines already do).

Continue previous example by grabbing a single block and a single automorphism. Apply the automorphism to the elements of the block and sort the result. Then discover the "new" block as an element of the list of blocks, as expected.

Careful: add one to argument, then subtract one from image, when applying automorphism to block. (Do you see why?)

2 Labelings of the Fano Plane

In how many ways? And how quickly can we figure this out? We have to work a

```
F = designs.ProjectiveGeometryDesign(2, 1, FiniteField(2))
F.is_block_design()
```

```
factorial(7)/F.automorphism_group().order()
```

Boom!