SYDE 710 Project

The Rubik's Cube Group

Samin Riasat

1 Introduction

The Rubik's Cube is one of the most popular puzzles in the world. It is a $3 \times 3 \times 3$ cube-shaped structure with each face consisting of 9 unit squares. To learn more about the puzzle and the history of the Cube, see [1].

The Rubik's Cube is mathematically interesting due to its many symmetries. Consequently, one may explore its symmetries using group theory. Our aim in this project is to explore some group-theoretic properties of the Rubik's Cube. In doing so we make extensive use of the computer algebra system `GAP` [3].

2 Generators

Following [4], we label the squares on the surface of the Rubik's Cube as in Figure 1. Let G denote the Rubik's Cube Group. Observe that G is generated by the 6 face-rotations.

```
gap> rot_u := ( 1, 3, 8, 6)( 2, 5, 7, 4)( 9,33,25,17)(10,34,26,18)(11,35,27,19);
(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19)
gap> rot_d := (41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40);
(14,22,30,38)(15,23,31,39)(16,24,32,40)(41,43,48,46)(42,45,47,44)
gap> rot_f := (17,19,24,22)(18,21,23,20)( 6,25,43,16)( 7,28,42,13)( 8,30,41,11);
(6,25,43,16)(7,28,42,13)(8,30,41,11)(17,19,24,22)(18,21,23,20)
gap> rot_b := (33,35,40,38)(34,37,39,36)( 3, 9,46,32)( 2,12,47,29)( 1,14,48,27);
(1,14,48,27)(2,12,47,29)(3,9,46,32)(33,35,40,38)(34,37,39,36)
gap> rot_l := ( 9,11,16,14)(10,13,15,12)( 1,17,41,40)( 4,20,44,37)( 6,22,46,35);
(1,17,41,40)(4,20,44,37)(6,22,46,35)(9,11,16,14)(10,13,15,12)
gap> rot_r := (25,27,32,30)(26,29,31,28)( 3,38,43,19)( 5,36,45,21)( 8,33,48,24);
(3,38,43,19)(5,36,45,21)(8,33,48,24)(25,27,32,30)(26,29,31,28)
gap> G:=Group(rot_u, rot_d, rot_l, rot_r, rot_f, rot_b);
<permutation group with 6 generators>
```

We can find the size of G.

```
gap> Size(G);
43252003274489856000
```

			1	2	3						
			4	U	5						
			6	7	8						
9	10	11	17	18	19	25	26	27	33	34	35
12	L	13	20	F	21	28	R	29	36	B	37
14	15	16	22	23	24	30	31	32	38	39	40
			41	42	43						
			44	D	45						
			46	47	48						

Figure 1: An Unfolded Rubik's Cube with Labelled Squares

3 Subgroups and Action

The subgroup generated by two "independent" generators is isomorphic to $C_4 \times C_4$.

```
gap> StructureDescription(Group(rot_u, rot_d));
"C4 x C4"
```

On the contrary, the subgroup generated by two "dependent" generators is much larger.

```
gap> Size(Group(rot_u, rot_l));
73483200
```

The subgroup generated by three such generators is even larger.

```
gap> H := Subgroup(G, [rot_u, rot_f, rot_l]);
<permutation group with 3 generators>
gap> Size(H);
170659735142400
```

G acts on the set X of the 48 squares by permutation. We can compute the orbits of the action (X, G) as follows. Note that the action of G on the squares is not transitive, since a corner square can move only to a corner square and vice versa. This is also shown by the code snippet below.

```
gap> Orbits(G);
[ [ 1, 3, 17, 14, 8, 38, 9, 41, 19, 48, 22, 6, 30, 33, 43, 11, 46,
  40, 24, 27, 25, 35, 16, 32 ],
  [ 2, 5, 12, 7, 36, 10, 47, 4, 28, 45, 34, 13, 29, 44, 20, 42, 26,
  21, 37, 15, 31, 18, 23, 39 ] ]

gap> stab1 := Stabiliser(G, 1);
<permutation group of size 1802166803103744000 with 5 generators>
gap> GeneratorsOfGroup(stab1);
[ (4,29,5,37)(8,22,38,24)(10,36,26,12)(16,32,43,19)(25,41,48,30),
  (3,11,43,38)(6,24,48,33)(7,31,23,12)(17,30,32,27)(18,45,42,37),
  (3,48,16,40)(5,47,13,7)(14,33,32,22)(18,26,39,20)(27,38,41,46),
  (4,5,36,12,18,23)(7,42,10,26,29,37)(8,30,16)(15,39)(19,24,41)(22,25,43)(44,47),
  (2,4)(10,34)(12,28)(21,37) ]
```

Note that if a group element fixes a square, it must fix the entire unit cube containing that square. Following standard terminology, we refer to each such unit cube as a *cubie*.

```
gap> stab9 := Stabiliser(G, 9);
<permutation group of size 1802166803103744000 with 19 generators>
gap> stab1 = stab9;
true
gap> stab35 := Stabiliser(G, 35);
<permutation group of size 1802166803103744000 with 19 generators>
gap> stab35 = stab9;
true
gap> stab2 := Stabiliser(G, 2);
<permutation group of size 1802166803103744000 with 19 generators>
gap> GeneratorsOfGroup(stab2);
[ (4,42,37)(10,23,12), (4,10)(12,37), (4,42)(5,37)(10,23)(12,26),
  (4,37)(5,18)(7,26)(10,12), (4,42,18)(5,37,29)(7,10,23)(12,36,26),
  (4,37,26,39,42)(5,47,23,10,12), (4,23,10,42)(5,7,21,12,26,18,28,37),
  (4,13,18,36,47,5)(7,29,39,26,10,20)(12,42)(23,37),
  (4,45,18,29,42,37,5)(7,36,23,12,26,10,31),
  (4,37,7,29,10,12,18,36)(5,26)(15,39)(44,47), (24,43,30)(32,38,48),
  (4,10)(12,42,37,23)(24,32)(30,38)(43,48), (16,22,41)(24,43,30),
  (4,26,42,18)(5,23,7,10)(16,38)(22,48)(32,41),
  (4,29,5,37)(8,22,38,24)(10,36,26,12)(16,32,43,19)(25,41,48,30),
  (3,43,16,32)(4,37,5,42)(10,12,26,23)(22,38,33,24)(27,30,41,48),
  (3,48,16,40)(5,47,13,7)(14,33,32,22)(18,26,39,20)(27,38,41,46),
  (3,11,43,38)(6,24,48,33)(7,31,23,12)(17,30,32,27)(18,45,42,37),
  (1,30,27,40)(3,14,9,43)(4,29,5,21,37,45,13,15)(6,25,41,48)(7,23)
  (8,22,38,11)(10,36,26,28,12,31,20,44)(16,32,17,19)(18,42)(24,33,46,35) ]
```

We can also consider the action of H above on X. One would expect that H is stabilises the 8 cubies in the opposite corner, which is indeed the case.

```
gap> H = Stabiliser(G, (29,31,32,36,38,45,47,48));
true
```

4 Composition Series

```
gap> series := CompositionSeries(G);
[ <permutation group of size 43252003274489856000 with 6 generators>,
  <permutation group of size 21626001637244928000 with 12 generators>,
  <permutation group of size 1072718335180800 with 9 generators>,
  <permutation group of size 357572778393600 with 8 generators>,
  <permutation group of size 119190926131200 with 7 generators>,
  <permutation group of size 39730308710400 with 6 generators>,
  <permutation group of size 13243436236800 with 5 generators>,
  <permutation group of size 4414478745600 with 4 generators>,
  <permutation group of size 1471492915200 with 3 generators>,
  <permutation group of size 490497638400 with 21 generators>,
  <permutation group of size 2048 with 11 generators>,
  <permutation group of size 1024 with 10 generators>,
  <permutation group of size 512 with 9 generators>,
  <permutation group of size 256 with 8 generators>,
  <permutation group of size 128 with 7 generators>,
  <permutation group of size 64 with 6 generators>,
  <permutation group of size 32 with 5 generators>,
  <permutation group of size 16 with 4 generators>,
  <permutation group of size 8 with 3 generators>,
  Group([ (4,10)(5,26), (2,34)(5,26) ]),
  Group([ (4,10)(5,26) ]), Group(()) ]
gap> Size(series);
22
```

So G has a composition series

$$G = G_0 \rhd \cdots \rhd G_n = \{()\} \qquad \text{(series)}$$

of length $n = 22$. We can use the following command to view the composition factors.

```
gap> DisplayCompositionSeries(G);
```

```
G (6 gens, size 4325200327448985600)
 | Z(2)
S (13 gens, size 2162600163724492800)
 | A(8) ~ A(3,2) = L(4,2) ~ D(3,2) = O+(6,2)
S (9 gens, size 1072718335180800)
 | Z(3)
S (8 gens, size 357572778393600)
 | Z(3)
S (7 gens, size 119190926131200)
 | Z(3)
S (6 gens, size 39730308710400)
 | Z(3)
S (5 gens, size 13243436236800)
 | Z(3)
S (4 gens, size 4414478745600)
 | Z(3)
S (3 gens, size 1471492915200)
 | Z(3)
S (21 gens, size 490497638400)
 | A(12)
S (11 gens, size 2048)
 | Z(2)
S (10 gens, size 1024)
 | Z(2)
S (9 gens, size 512)
 | Z(2)
S (8 gens, size 256)
 | Z(2)
S (7 gens, size 128)
 | Z(2)
S (6 gens, size 64)
 | Z(2)
S (5 gens, size 32)
 | Z(2)
S (4 gens, size 16)
 | Z(2)
S (3 gens, size 8)
 | Z(2)
S (2 gens, size 4)
 | Z(2)
S (1 gens, size 2)
 | Z(2)
1 (0 gens, size 1)
```

We can understand the component actions $(G_i/G_{i+1}, G_i/G_{i+1})$ of (series) as follows.

```
gap> i := 1;
1
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (1,27,24,40)(2,13,39,31,29,26,37,34,20,47,45,36,5,12)(3,30,14,9)
    (4,44,18)(6,32)(7,10,15)(11,48)(17,38)(23,42)(33,43,46,35),
  (1,6,24)(2,45,42,15)(3,14,38)(4,7,36,26,20,39,12,21)
    (5,13,47,37,28,10,18,29)(9,17,30)(11,43,35)(23,44,34,31)(27,40,32)(33,46,48),
  (3,38)(4,21,13,44,47,7)(5,26)(8,24,16,40)(10,28,20,15,39,18)(14,25,30,22)
    (19,43,41,46)(23,29,42,36)(27,32)(33,48),
  (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (4,42,10,23)(5,37,28,18,26,12,21,7)(24,43,30)(32,38,48),
  (4,36,18,12,10,29,7,37)(5,26)(15,39)(16,22,41)(24,43,30)(44,47),
  (4,29,37,28,10,36,12,21)(5,18,26,7)(8,19,25)(24,30,43),
  (3,27,33)(4,37,29,26,28,13,7,23)(5,21,20,18,42,10,12,36)(15,39)(24,30,43)(44,47),
  (4,47,26,21,18,36,20,10,39,5,28,7,29,13)(14,46,40)(23,42)(24,43,30),
  (4,13,7,23,37,5,28,31,10,20,18,42,12,26,21,45)(6,17,11)(15,39,44,47)(24,43,30),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (1,6,27,40)(2,45,13,15)(3,14,9,17)(11,33,46,35)(20,44,34,31) ]
  -> [ (), (), (), (), (), (), (), (), (), (), (), (), (1,2) ]
2
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (1,27,24,40)(2,13,39,31,29,26,37,34,20,47,45,36,5,12)(3,30,14,9)
    (4,44,18)(6,32)(7,10,15)(11,48)(17,38)(23,42)(33,43,46,35),
  (1,6,24)(2,45,42,15)(3,14,38)(4,7,36,26,20,39,12,21)(5,13,47,37,28,10,18,29)
    (9,17,30)(11,43,35)(23,44,34,31)(27,40,32)(33,46,48),
  (3,38)(4,21,13,44,47,7)(5,26)(8,24,16,40)(10,28,20,15,39,18)(14,25,30,22)
    (19,43,41,46)(23,29,42,36)(27,32)(33,48),
  (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (4,42,10,23)(5,37,28,18,26,12,21,7)(24,43,30)(32,38,48),
  (4,36,18,12,10,29,7,37)(5,26)(15,39)(16,22,41)(24,43,30)(44,47),
  (4,29,37,28,10,36,12,21)(5,18,26,7)(8,19,25)(24,30,43),
  (3,27,33)(4,37,29,26,28,13,7,23)(5,21,20,18,42,10,12,36)(15,39)(24,30,43)(44,47),
  (4,47,26,21,18,36,20,10,39,5,28,7,29,13)(14,46,40)(23,42)(24,43,30),
  (4,13,7,23,37,5,28,31,10,20,18,42,12,26,21,45)(6,17,11)(15,39,44,47)(24,43,30),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30) ]
  -> [ (1,4,8,3)(2,7), (1,2,8)(3,7,4), (3,5,8,6)(4,7), (), (), (), (), (), (), (),
    (), () ]
```

```
3
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (4,36,18,12,10,29,7,37)(5,26)(15,39)(16,22,41)(24,43,30)(44,47),
  (4,29,37,28,10,36,12,21)(5,18,26,7)(8,19,25)(24,30,43),
  (3,27,33)(4,37,29,26,28,13,7,23)(5,21,20,18,42,10,12,36)(15,39)(24,30,43)(44,47),
  (4,47,26,21,18,36,20,10,39,5,28,7,29,13)(14,46,40)(23,42)(24,43,30),
  (4,13,7,23,37,5,28,31,10,20,18,42,12,26,21,45)(6,17,11)(15,39,44,47)(24,43,30),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (4,42,10,23)(5,37,28,18,26,12,21,7)(24,43,30)(32,38,48) ]
  -> [ (), (), (), (), (), (), (), (), (1,2,3) ]
4
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (1,35,9)(2,28,23,31,12,36,18,5,39,13,4)(6,17,11)(7,26,47,20,10,34,21,42,45,37,29),
  (1,9,35)(2,44,4,45,13,42,7,47,12,5)(6,17,11)(10,31,20,23,18,39,37,26,34,15)
    (14,46,40)(21,36)(28,29),
  (1,35,9)(2,28,45,18,10,34,21,31,7,4)(3,27,33)
    (5,42,37,39,13,44,29,26,23,12,47,20,15,36)(6,11,17),
  (1,35,9)(2,28,23,10,44,20,18,31,12,39,26)
    (4,15,13,7,45,37,47,5,34,21,42)(6,17,11)(8,19,25)(24,30,43),
  (4,36,18,12,10,29,7,37)(5,26)(15,39)(16,22,41)(24,43,30)(44,47) ]
  -> [ (), (), (), (), (), (), (), (1,2,3) ]
5
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (1,35,9)(2,28,23,31,12,36,18,5,39,13,4)(6,17,11)(7,26,47,20,10,34,21,42,45,37,29),
  (1,9,35)(2,44,4,45,13,42,7,47,12,5)(6,17,11)(10,31,20,23,18,39,37,26,34,15)
    (14,46,40)(21,36)(28,29),
  (1,35,9)(2,28,45,18,10,34,21,31,7,4)(3,27,33)
    (5,42,37,39,13,44,29,26,23,12,47,20,15,36)(6,11,17),
  (1,35,9)(2,28,23,10,44,20,18,31,12,39,26)
    (4,15,13,7,45,37,47,5,34,21,42)(6,17,11)(8,19,25)(24,30,43) ]
  -> [ (), (), (), (), (), (), (1,2,3) ]
```

```
6
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (1,35,9)(2,28,23,31,12,36,18,5,39,13,4)(6,17,11)(7,26,47,20,10,34,21,42,45,37,29),
  (1,9,35)(2,44,4,45,13,42,7,47,12,5)(6,17,11)(10,31,20,23,18,39,37,26,34,15)
    (14,46,40)(21,36)(28,29),
  (1,35,9)(2,28,45,18,10,34,21,31,7,4)(3,27,33)
    (5,42,37,39,13,44,29,26,23,12,47,20,15,36)(6,11,17) ]
  -> [ (), (), (), (), (), (1,2,3) ]
7
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (1,35,9)(2,28,23,31,12,36,18,5,39,13,4)(6,17,11)(7,26,47,20,10,34,21,42,45,37,29),
  (1,9,35)(2,44,4,45,13,42,7,47,12,5)(6,17,11)(10,31,20,23,18,39,37,26,34,15)
    (14,46,40)(21,36)(28,29) ]
  -> [ (), (), (), (), (1,2,3) ]
8
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
(2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30),
  (1,35,9)(2,28,23,31,12,36,18,5,39,13,4)(6,17,11)(7,26,47,20,10,34,21,42,45,37,29) ]
  -> [ (), (), (), (1,2,3) ]
9
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,37,15,21,42,34,12,44,28,23)(4,36,26)(5,10,29)(7,18)(13,31,47)(20,45,39),
  (2,44,42,20,21,12,18)(4,47,5,36,45)(7,34,15,23,13,28,37)(10,39,26,29,31),
  (1,9,35)(2,44,31,12,39,34,15,45,37,47)(4,23,29,26)(5,10,42,36)(7,21,18,28)
    (24,43,30) ]
  -> [ (), (), (1,2,3) ]
10
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (2,5)(4,21)(10,28)(26,34), (4,5,7)(10,26,18), (4,20)(5,7)(10,13)(18,26),
  (4,15,20)(5,7,37)(10,44,13)(12,26,18),
  (4,7)(5,36,37,15,42,20)(10,18)(12,44,23,13,26,29),
```

```
  (4,37,15,5)(10,12,44,26)(23,39)(42,47),
  (4,20,37,21)(5,7)(10,13,12,28)(18,26),
  (4,20,42,7,5)(10,13,23,18,26),
  (4,20,5,45,37,15,7)(10,13,26,31,12,44,18),
  (4,5)(10,26)(12,13)(20,37), (4,10)(7,18), (4,10)(13,20), (4,10)(12,37),
  (4,10)(15,44), (4,10)(23,42), (4,10)(21,28), (4,10)(29,36),
  (4,10)(31,45), (4,10)(39,47), (2,34)(4,10), (4,10)(5,26) ]
  -> [ (1,11)(5,12), (10,12,11), (9,12)(10,11), (7,9,12)(8,11,10),
       (4,8,7,6,9,11)(10,12), (2,6)(7,11,12,8), (5,12,9,8)(10,11),
       (6,10,11,12,9), (3,8,7,10,12,9,11), (8,9)(11,12), (), (), (), (),
       (), (), (), (), (), (), () ]
11
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(13,20), (4,10)(12,37), (4,10)(15,44), (4,10)(23,42), (4,10)(21,28),
  (4,10)(29,36), (4,10)(31,45), (4,10)(39,47), (2,34)(4,10), (4,10)(5,26),
  (2,34)(4,10)(5,26)(7,18)(31,45)(39,47) ]
  -> [ (), (), (), (), (), (), (), (), (), (), (1,2) ]
12
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(12,37), (4,10)(15,44), (4,10)(23,42), (4,10)(21,28), (4,10)(29,36),
  (4,10)(31,45), (4,10)(39,47), (2,34)(4,10), (4,10)(5,26), (13,20)(15,44) ]
  -> [ (), (), (), (), (), (), (), (), (), (1,2) ]
13
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(15,44), (4,10)(23,42), (4,10)(21,28), (4,10)(29,36), (4,10)(31,45),
  (4,10)(39,47), (2,34)(4,10), (4,10)(5,26), (12,37)(39,47) ]
  -> [ (), (), (), (), (), (), (), (), (1,2) ]
14
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(23,42), (4,10)(21,28), (4,10)(29,36), (4,10)(31,45), (4,10)(39,47),
  (2,34)(4,10), (4,10)(5,26), (2,34)(4,10)(15,44)(31,45) ]
  -> [ (), (), (), (), (), (), (), (1,2) ]
15
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(21,28), (4,10)(29,36), (4,10)(31,45), (4,10)(39,47), (2,34)(4,10),
  (4,10)(5,26), (23,42)(39,47) ]
  -> [ (), (), (), (), (), (), (1,2) ]
16
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(5,26), (2,34)(4,10), (5,26)(39,47), (31,45)(39,47), (5,26)(29,36),
  (2,34)(21,28) ]
  -> [ (), (), (), (), (), (1,2) ]
```

```
17
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(5,26), (2,34)(4,10), (5,26)(39,47), (31,45)(39,47), (5,26)(29,36) ]
  -> [ (), (), (), (), (1,2) ]
18
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(5,26), (2,34)(4,10), (5,26)(39,47), (31,45)(39,47) ]
  -> [ (), (), (), (1,2) ]
19
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(5,26), (2,34)(4,10), (5,26)(39,47) ] -> [ (), (), (1,2) ]
20
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
[ (4,10)(5,26), (2,34)(4,10) ] -> [ (), (1,2) ]
21
gap> NaturalHomomorphismByNormalSubgroup(series[i], series[i+1]); i := i+1;
IdentityMapping( Group([ (4,10)(5,26) ]) )
22
```

This shows that G embeds in a wreath product of the form

$$\mathbb{Z}_2 \wr A_8 \wr \mathbb{Z}_3 \wr \mathbb{Z}_3 \wr \mathbb{Z}_3 \wr \mathbb{Z}_3 \wr \mathbb{Z}_3 \wr \mathbb{Z}_3 \wr \mathbb{Z}_3 \wr A_{12} \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2 \wr \mathbb{Z}_2$$

with the component actions described as above.

Next, we obtain coset representatives for (series) as follows.

```
gap> RightCosets(series[1], series[2]);
[ RightCoset(<permutation group of size 21626001637244928000 with 11 generators>,
  ()),
  RightCoset(<permutation group of size 21626001637244928000 with 11 generators>,
  (24,32)(30,38)(31,39)(43,48)(45,47)) ]
```

The next quotient is isomorphic to A_8, whence there are 20160 coset representatives. Since the list is too big so we do not display the generated output here.

```
gap> RightCosets(series[2], series[3]);
```

And we can keep going.

```
gap> RightCosets(series[3], series[4]);
[ RightCoset(<permutation group of size 357572778393600 with 8 generators>,
    ()),
  RightCoset(<permutation group of size 357572778393600 with 8 generators>,
    (4,23,10,42)(5,7,21,12,26,18,28,37)(24,30,43)(32,48,38)),
  RightCoset(<permutation group of size 357572778393600 with 8 generators>,
```

```
    (4,10)(5,21,26,28)(7,12,18,37)(23,42)(24,43,30)(32,38,48)) ]
gap> RightCosets(series[4], series[5]);
[ RightCoset(<permutation group of size 119190926131200 with 7 generators>,
    ()),
  RightCoset(<permutation group of size 119190926131200 with 7 generators>,
    (4,37,7,29,10,12,18,36)(5,26)(15,39)(16,41,22)(24,30,43)(44,47)),
  RightCoset(<permutation group of size 119190926131200 with 7 generators>,
    (4,7,10,18)(12,36,37,29)(16,22,41)(24,43,30)) ]
gap> RightCosets(series[5], series[6]);
[ RightCoset(<permutation group of size 39730308710400 with 6 generators>,
    ()),
  RightCoset(<permutation group of size 39730308710400 with 6 generators>,
    (1,9,35)(2,26,39,12,31,18,20,44,10,23,28)(4,42,21,34,5,47,37,45,7,13,15)
      (6,11,17)(8,25,19)(24,43,30)),
  RightCoset(<permutation group of size 39730308710400 with 6 generators>,
    (1,35,9)(2,39,31,20,10,28,26,12,18,44,23)(4,21,5,37,7,15,42,34,47,45,13)
      (6,17,11)(8,19,25)(24,30,43)) ]
gap> RightCosets(series[6], series[7]);
[ RightCoset(<permutation group of size 13243436236800 with 5 generators>,
    ()),
  RightCoset(<permutation group of size 13243436236800 with 5 generators>,
    (1,9,35)(2,4,7,31,21,34,10,18,45,28)(3,33,27)
      (5,36,15,20,47,12,23,26,29,44,13,39,37,42)(6,17,11)),
  RightCoset(<permutation group of size 13243436236800 with 5 generators>,
    (1,35,9)(2,7,21,10,45)(3,27,33)(4,31,34,18,28)(5,15,47,23,29,13,37)
      (6,11,17)(12,26,44,39,42,36,20)) ]
gap> RightCosets(series[7], series[8]);
[ RightCoset(<permutation group of size 4414478745600 with 4 generators>,
    ()),
  RightCoset(<permutation group of size 4414478745600 with 4 generators>,
    (1,35,9)(2,5,12,47,7,42,13,45,4,44)(6,11,17)
      (10,15,34,26,37,39,18,23,20,31)(14,40,46)(21,36)(28,29)),
  RightCoset(<permutation group of size 4414478745600 with 4 generators>,
    (1,9,35)(2,12,7,13,4)(5,47,42,45,44)(6,17,11)
      (10,34,37,18,20)(14,46,40)(15,26,39,23,31)) ]
gap> RightCosets(series[8], series[9]);
[ RightCoset(<permutation group of size 1471492915200 with 3 generators>,
    ()),
  RightCoset(<permutation group of size 1471492915200 with 3 generators>,
    (1,9,35)(2,4,13,39,5,18,36,12,31,23,28)(6,11,17)
      (7,29,37,45,42,21,34,10,20,47,26)),
  RightCoset(<permutation group of size 1471492915200 with 3 generators>,
```

```
    (1,35,9)(2,13,5,36,31,28,4,39,18,12,23)(6,17,11)
      (7,37,42,34,20,26,29,45,21,10,47)) ]
gap> RightCosets(series[9], series[10]);
[ RightCoset(<permutation group of size 490497638400 with 21 generators>,
    ()),
  RightCoset(<permutation group of size 490497638400 with 21 generators>,
    (1,35,9)(2,47,37,45,15,34,39,12,31,44)(4,26,29,23)
      (5,36,42,10)(7,28,18,21)(24,30,43)),
  RightCoset(<permutation group of size 490497638400 with 21 generators>,
    (1,9,35)(2,37,15,39,31)(4,29)(5,42)(7,18)(10,36)(12,44,47,45,34)(21,28)
      (23,26)(24,43,30)) ]
```

The next factor is isomorphic to A_{12}, whence there are 239500800 coset representatives. Since the list is too big so we do not display the generated output here.

```
gap> RightCosets(series[10], series[11]);
```

And now we can complete the rest of the list.

```
gap> RightCosets(series[11], series[12]);
[ RightCoset(<permutation group of size 1024 with 10 generators>,()),
  RightCoset(<permutation group of size 1024 with 10 generators>,(7,18)(39,47)) ]
gap> RightCosets(series[12], series[13]);
[ RightCoset(<permutation group of size 512 with 9 generators>,()),
  RightCoset(<permutation group of size 512 with 9 generators>,(13,20)(39,47)) ]
gap> RightCosets(series[13], series[14]);
[ RightCoset(<permutation group of size 256 with 8 generators>,()),
  RightCoset(<permutation group of size 256 with 8 generators>,(12,37)(39,47)) ]
gap> RightCosets(series[14], series[15]);
[ RightCoset(<permutation group of size 128 with 7 generators>,()),
  RightCoset(<permutation group of size 128 with 7 generators>,(15,44)(39,47)) ]
gap> RightCosets(series[15], series[16]);
[ RightCoset(<permutation group of size 64 with 6 generators>,()),
  RightCoset(<permutation group of size 64 with 6 generators>,(23,42)(39,47)) ]
gap> RightCosets(series[16], series[17]);
[ RightCoset(<permutation group of size 32 with 5 generators>,()),
  RightCoset(<permutation group of size 32 with 5 generators>,(21,28)(39,47)) ]
gap> RightCosets(series[17], series[18]);
[ RightCoset(<permutation group of size 16 with 4 generators>,()),
  RightCoset(<permutation group of size 16 with 4 generators>,(29,36)(39,47)) ]
gap> RightCosets(series[18], series[19]);
[ RightCoset(<permutation group of size 8 with 3 generators>,()),
  RightCoset(<permutation group of size 8 with 3 generators>,(31,45)(39,47)) ]
gap> RightCosets(series[19], series[20]);
```

```
[ RightCoset(Group([ (4,10)(5,26), (2,34)(4,10) ]),()),
  RightCoset(Group([ (4,10)(5,26), (2,34)(4,10) ]),(5,26)(39,47)) ]
gap> RightCosets(series[20], series[21]);
[ RightCoset(Group([ (4,10)(5,26) ]),()),
  RightCoset(Group([ (4,10)(5,26) ]),(2,34)(5,26)) ]
gap> RightCosets(series[21], series[22]);
[ RightCoset(Group(()),()), RightCoset(Group(()),(4,10)(5,26)) ]
```

The coset representatives above give coarse-to-fine coordinates on G.

5 Expressing Coset Representatives Using Generators

```
gap> hom := EpimorphismFromFreeGroup(G:names:=["u","d","l","r","f","b"]);
[ u, d, l, r, f, b ] ->
  [ (1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19),
  (14,22,30,38)(15,23,31,39)(16,24,32,40)(41,43,48,46)(42,45,47,44),
  (1,17,41,40)(4,20,44,37)(6,22,46,35)(9,11,16,14)(10,13,15,12),
  (3,38,43,19)(5,36,45,21)(8,33,48,24)(25,27,32,30)(26,29,31,28),
  (6,25,43,16)(7,28,42,13)(8,30,41,11)(17,19,24,22)(18,21,23,20),
  (1,14,48,27)(2,12,47,29)(3,9,46,32)(33,35,40,38)(34,37,39,36) ]
gap> PreImagesRepresentative(hom,(4,10)(5,26));
u*f*r*u*r^-1*u^-1*f^-1*u^-1*l^-1*u^-1*b^-1*u^2*b*l*f*r*u^-1*r^-1*f^-1
gap> PreImagesRepresentative(hom,(2,34)(5,26));
l^-1*u^-1*b^-1*u^2*b*l*f*r*u^-1*r^-1*f^-1*u*f*r*u*r^-1*u^-1*f^-1*u^-1
gap> PreImagesRepresentative(hom,(5,26)(39,47));
u*f*r*u*r^-1*u^-1*f^-1*u^-1*l^-1*u^-1*b^-1*u*b*l*u^-1*f*u*l*u*l^-1*f^-1*l*
u^-1*l^-1*f*(u^-1*f^-1)^2*u^-1*l*u*l^-1*f*l*u^-1*l^-1*u^-1*f^-1*l*f*l^-1*
u^-1*l^-1*u*l*u^-1*f*u*r*u^-1*r^-1*f^-1*u*f*u*f^-1*u^-1*l^-1*u^-1*l*u*l^-1*
u*l*u^-1*l^-1*f^-1*l^-1*f*l^2*u*l^2*u^-1*l^2*b^-1*u^-1*b*l*d*r^-2*d^-1*b*
d^-1*b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(31,45)(39,47));
l^-1*u^-1*l*f*r*u*r^-1*f^-1*l*u*f*u^-1*f^-1*l^-1*u^2*l*u*l^-1*f*u^-1*f^-1*
u*l*u^-1*l^-1*u*f^-1*l*f*l^-1*u*l*f*u*f^-1*u^-1*l^-1*u^2*f^-1*l^-1*u^-1*
(l*u*f)^2*u^-1*f^-1*l^-1*u^-1*l^-1*u*l*u^-1*l*f*u*f^-1*u^-1*l^-1*u^-2*l^-1*
f^-1*l^-2*f*l*b^-1*u*b*l^-2*u^-1*f*l*f*l^-1*f^-1*u^-2*r^-1*f^-1*r*u*f*l^-1*
u*l^-1*d^-1*b*d*u^-2*b^-1*r*b*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(29,36)(39,47));
l^-1*u^-1*l*f*r*u*r^-1*f^-1*l*u*f*u^-1*f^-1*l^-1*u^2*l*u*l^-1*f*u^-1*f^-1*
u*l*u^-1*l^-1*u*f^-1*l*f*l^-1*u*l*f*u*f^-1*u^-1*l^-1*u^2*f^-1*l^-1*u^-1*
(l*u*f)^2*u^-1*f^-1*l^-1*u^-1*l^-1*u*l*u^-1*l*f*u*f^-1*u^-1*l^-1*u^-2*l^-1*
f^-1*l^-2*f*l*b^-1*u*b*l^-2*u^-1*f*l*f*l^-1*f^-1*u^-2*r^-1*f^-1*r*u*f*l^-1*
u*l^-1*d^-1*b*d*u^-2*b^-1*r*b*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(21,28)(39,47));
```

```
l^-1*b^-1*u^-1*b*u*l*u*f*u*r*u^-1*r^-1*f^-1*u*l*u*l^-1*f*u^-1*f^-1*u*l*u^-1*
l^-1*u*f^-1*l*f*l^-1*u*l*f*u*f^-1*u^-1*l^-1*u*l*u*f*u^-1*f^-2*l^-1*f*l^-1*
u^-1*l*f*u*f^-1*u*l*u*(l*b*l*b^-1)^2*u^-1*d*f^-1*d^-1*u^-2*d^-1*l*d*f^-1*
l^-1*f^-1*d^-1*b^-2*d*b*d^2*b^-1*l^-1*r^-1*u^-2*(f^-1*r^-1)^2*d*r^-2*d^-1*
b*d^-1*b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(23,42)(39,47));
u*f*r*u*r^-1*u^-1*f^-1*u^-1*(l^-1*u^-1*b^-1*u^2*b*l*f*r*u^-1*r^-1*f^-1)^2*u*
f*r*u*r^-1*u^-1*f^-1*u^-1*l^-1*u^-1*l^2*f^-1*l^-1*f*l^-1*u*l*f*u^-1*r*u*r^-1*
u^-1*f^-1*u^-1*f*u*f^-1*u^-1*l^-1*u^-1*l*u^-1*l^-1*u*b^-1*u*b*f^-1*l^-1*f*l*
u*b^-1*u*b*l*u^-1*l^-1*b*l*b^-1*d*f*d^-1*l*r^-1*f*r*(f*l^-1*r*u^-1*r^-1*l^-1)^2*
d*r^-2*d^-1*b*d^-1*b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(15,44)(39,47));
l^-1*b^-1*u^-1*b*u*l*u*f*u*r*u^-1*r^-1*f^-1*u*l*u*l^-1*f*u^-1*f^-1*u*l*u^-1*
l^-1*u*f^-1*l*f*u*f*u^-1*f^-1*l^-2*u^-1*b^-1*u*b*l*u*f*u*r*u^-1*r^-1*f^-1*u*
f*u*f^-1*u^-1*l^-1*u^-1*l*u*l^-1*u*b^-1*u*b*l*u*l^-1*f^-1*l^-1*f*l*u^-1*b^-1*
u*b*l^-1*d*r^-2*d^-1*b*d^-1*b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(12,37)(39,47));
u*f*r*u*r^-1*u^-1*f^-1*u^-1*l^-1*u^-1*b^-1*u^2*b*l*f*r*u^-1*r^-1*f^-1*l^-1*
b^-1*u^-1*b*u*l*u*f*u*r*u^-1*r^-1*f^-1*u^-2*f*u*l*u*l^-1*f^-1*l*u^-1*l^-1*f*
u^-1*f^-1*u^-1*l*u*l^-1*u*l*u^-2*l^-1*f^-1*l*f*l^-1*u^-1*l^-1*u*l*u^-1*f*u*
r*u^-1*r^-1*u^-1*f^-1*l^-1*u^-1*f^-1*l*f*l*u^-1*l*u*l^2*u^-1*l^2*b^-1*u^-1*
b*l*d*r^-2*d^-1*b*d^-1*b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(13,20)(39,47));
l^-1*u^-1*l*f*r*u*r^-1*f^-1*l*u*f*u^-1*f^-1*l^-1*u^-1*f*u*l*u*l^-1*f^-1*l*
u^-1*l^-1*f*u^-1*f^-1*u^-1*l*u^-1*l^-1*u*f^-1*u^-1*l*u*l^-1*f*u*l^-1*u^-1*
l^-1*b*l*b^-1*u*l*u^-2*l*f*u*f^-1*u^-1*l^-2*u*l*u*f*u^-1*f^-1*u^-2*l^-1*u*
l*u^-1*l^-1*f^-1*l^-1*f*l^2*u*l^2*u^-1*l^2*b^-1*u^-1*b*l*d*r^-2*d^-1*b*d^-1*
b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
gap> PreImagesRepresentative(hom,(7,18)(39,47));
u*f*r*u*r^-1*u^-1*f^-1*u^-1*l^-1*u^-1*b^-1*u*b*l*u^-1*f^-1*l*u^-1*l^-1*u*
f*u^-1*l*(u*l^-1)^2*u^-1*l^-1*b*l*b^-1*u*l*u^-1*f*u*r*u^-1*r^-1*f^-1*u*f*
u*f^-1*u^-1*l^-1*u^-1*l*u*l^-1*u*l*u^-1*l^-1*f^-1*l^-1*f*l^2*u*l^2*u^-1*l^2*
b^-1*u^-1*b*l*d*r^-2*d^-1*b*d^-1*b^-1*f^-1*r^-2*u^-2*l^-1*b^-1*l^-1*d^-1
```

We stop here since the next quotient is isomorphic to A_{12}.

6 Solving by Levels

With a full list of coset representatives we can express any $g \in G$ in the Frobenius-Lagrange coordinate representation. This will allow us to express any valid configuration of the Rubik's Cube in terms of the coset representatives. In order to solve the Cube we can then simply perform the corresponding inverse rotations in reverse order.

7 Macros for Special Permutations

Since the coset representatives above may be used to obtain any valid permutation of the squares, we can use them to define macros to achieve certain permutations. Here are some examples.

7.1 Flip Two Opposite Edge Cubies on the Same Face

This corresponds to a permutation equivalent to $(4,10)(5,26)$, which is given by the following macro.

$$\texttt{ufrur}^{-1}\texttt{u}^{-1}\texttt{f}^{-1}\texttt{u}^{-1}\texttt{l}^{-1}\texttt{u}^{-1}\texttt{b}^{-1}\texttt{u}^{2}\texttt{blfru}^{-1}\texttt{r}^{-1}\texttt{f}^{-1}$$

7.2 Flip Two Adjacent Edge Cubies on the Same Face

This corresponds to a permutation equivalent to $(2,34)(5,26)$, which is given by the following macro.

$$\texttt{l}^{-1}\texttt{u}^{-1}\texttt{b}^{-1}\texttt{u}^{2}\texttt{blfru}^{-1}\texttt{r}^{-1}\texttt{f}^{-1}\texttt{ufrur}^{-1}\texttt{u}^{-1}\texttt{f}^{-1}\texttt{u}^{-1}$$

7.3 Flip Two Edge Cubies on Adjacent Faces

This corresponds to a permutation equivalent to $(5,26)(39,47)$, which is given by the following macro.

$\texttt{ufrur}^{-1}\texttt{u}^{-1}\texttt{f}^{-1}\texttt{u}^{-1}\texttt{l}^{-1}\texttt{u}^{-1}\texttt{b}^{-1}\texttt{ublu}^{-1}\texttt{fulul}^{-1}\texttt{f}^{-1}\texttt{lu}^{-1}\texttt{l}^{-1}\texttt{f}(\texttt{u}^{-1}\texttt{f}^{-1})^{2}\texttt{u}^{-1}\texttt{lul}^{-1}\texttt{flu}^{-1}\texttt{l}^{-1}\texttt{u}^{-1}\texttt{f}^{-1}$
$\texttt{lfl}^{-1}\texttt{u}^{-1}\texttt{l}^{-1}\texttt{ulu}^{-1}\texttt{furu}^{-1}\texttt{r}^{-1}\texttt{f}^{-1}\texttt{ufuf}^{-1}\texttt{u}^{-1}\texttt{l}^{-1}\texttt{u}^{-1}\texttt{lul}^{-1}\texttt{ulu}^{-1}\texttt{l}^{-1}\texttt{f}^{-1}\texttt{l}^{-1}\texttt{fl}^{2}\texttt{ul}^{2}\texttt{u}^{-1}\texttt{l}^{2}\texttt{b}^{-1}\texttt{u}^{-1}\texttt{b}$
$\texttt{ldr}^{-2}\texttt{d}^{-1}\texttt{bd}^{-1}\texttt{b}^{-1}\texttt{f}^{-1}\texttt{r}^{-2}\texttt{u}^{-2}\texttt{l}^{-1}\texttt{b}^{-1}\texttt{l}^{-1}\texttt{d}^{-1}$

7.4 Swap Opposite Pairs of Edge Cubies on a Face

This corresponds to a permutation equivalent to $(4,5)(10,26)(2,7)(18,34)$, which is given by the following macro.

$$\texttt{lfuf}^{-1}\texttt{u}^{-1}\texttt{l}^{-1}\texttt{fru}^{-1}\texttt{r}^{-1}\texttt{f}^{-1}\texttt{l}^{-1}\texttt{ulu}^{-1}\texttt{lufu}^{-1}\texttt{f}^{-1}\texttt{l}^{-1}\texttt{uf}^{-1}\texttt{l}^{-1}\texttt{u}^{-1}\texttt{luf}$$

7.5 Swap Opposite Pairs of Corner Cubies on a Face

This corresponds to a permutation equivalent to $(1,8)(9,25)(35,19)(3,6)(11,27)(17,33)$, which is given by the following macro.

$$\texttt{lfuf}^{-1}\texttt{u}^{-1}\texttt{l}^{-1}\texttt{fru}^{-1}\texttt{r}^{-1}\texttt{f}^{-1}\texttt{l}^{-1}\texttt{ulu}^{-1}\texttt{lufu}^{-1}\texttt{f}^{-1}\texttt{l}^{-1}\texttt{uf}^{-1}\texttt{l}^{-1}\texttt{u}^{-1}\texttt{lufu}^{-2}$$

7.6 Flip Two Diametrically Opposite Edge Cubies

This corresponds to a permutation equivalent to $(7,18)(39,47)$, which is given by the following macro.

$$\mathtt{ufrur^{-1}u^{-1}f^{-1}u^{-1}l^{-1}u^{-1}b^{-1}ublu^{-1}f^{-1}lu^{-1}l^{-1}ufu^{-1}l(ul^{-1})^{2}u^{-1}l^{-1}blb^{-1}ulu^{-1}furu^{-1}r^{-1}}$$
$$\mathtt{f^{-1}ufuf^{-1}u^{-1}l^{-1}u^{-1}lul^{-1}ulu^{-1}l^{-1}f^{-1}l^{-1}fl^{2}ul^{2}u^{-1}l^{2}b^{-1}u^{-1}bldr^{-2}d^{-1}bd^{-1}b^{-1}f^{-1}r^{-2}u^{-2}}$$
$$\mathtt{l^{-1}b^{-1}l^{-1}d^{-1}}$$

7.7 Rotate Two Opposite Corner Cubies on the Same Face in Same Orientation

This corresponds to a permutation equivalent to $(1,9,35)(8,25,19)$, which is given by the following macro.

$$\mathtt{lufu^{-1}f^{-1}l^{-1}u(ufuru^{-1}r^{-1}f^{-1})^{2}u^{-1}(frur^{-1}f^{-1}l^{-1}b^{-1}u^{-2}bul)^{2}ufuru^{-1}r^{-1}f^{-1}u^{-1}}$$

7.8 Rotate Two Opposite Corner Cubies on the Same Face in Reverse Orientation

This corresponds to a permutation equivalent to $(1,9,35)(8,19,25)$, which is given by the following macro.

$$\mathtt{ufrur^{-1}u^{-1}f^{-1}u^{-1}l^{-1}u^{-1}b^{-1}u^{2}blfru^{-1}r^{-1}f^{-1}lfuf^{-1}u^{-1}l^{-1}fru^{-1}r^{-1}f^{-1}l^{-1}ulu^{-1}lf^{-1}l^{-1}f}$$
$$\mathtt{u^{-1}lul^{-1}u^{-1}fuf^{-1}lu^{-1}l^{-1}u^{-2}frur^{-1}u^{-1}f^{-1}u^{-2}lfuf^{-1}u^{-1}l^{-1}}$$

7.9 Rotate Two Diametrically Opposite Corner Cubies on the Same Face in Same Orientation

This corresponds to a permutation equivalent to $(1,9,35)(24,30,43)$, which is given by the following macro.

$$\mathtt{ufrur^{-1}u^{-1}f^{-1}u^{-1}(l^{-1}u^{-1}b^{-1}u^{2}blfru^{-1}r^{-1}f^{-1})^{2}ufrur^{-1}u^{-1}f^{-1}u^{-2}lul^{-1}ulu^{-2}l^{-1}f^{-1}lfl^{-1}}$$
$$\mathtt{u^{-1}l^{-1}ulu^{-1}lufu^{-1}f^{-1}l^{-1}u^{-2}l^{-1}ulu^{-1}lfuf^{-1}u^{-1}l^{-1}u^{-2}l^{-1}b^{-1}ub(lu)^{2}flfl^{-1}f^{-1}u^{-1}r^{-1}f^{-2}}$$
$$\mathtt{rfl^{-1}u^{-1}f}$$

7.10 Rotate Two Diametrically Opposite Corner Cubies in Reverse Orientation

This corresponds to a permutation equivalent to $(1,9,35)(24,43,30)$, which is given by the following macro.

$\mathtt{f}^{-1}\mathtt{ulf}^{-1}\mathtt{r}^{-1}\mathtt{f}^{2}\mathtt{ruflf}^{-1}\mathtt{l}^{-1}\mathtt{f}^{-1}(\mathtt{u}^{-1}\mathtt{l}^{-1})^{2}\mathtt{b}^{-1}\mathtt{u}^{-1}\mathtt{blu}^{2}\mathtt{lufu}^{-1}\mathtt{f}^{-1}\mathtt{l}^{-1}\mathtt{ul}^{-1}\mathtt{u}^{-1}\mathtt{lu}^{2}\mathtt{lfuf}^{-1}\mathtt{u}^{-1}\mathtt{l}^{-1}\mathtt{u}$
$\mathtt{l}^{-1}\mathtt{u}^{-1}\mathtt{lulf}^{-1}\mathtt{l}^{-1}\mathtt{flu}^{2}\mathtt{l}^{-1}\mathtt{u}^{-1}\mathtt{lu}^{-1}\mathtt{f}^{-1}\mathtt{l}^{-1}\mathtt{fu}^{-1}\mathtt{lul}^{-1}\mathtt{u}^{-1}\mathtt{fuf}^{-1}\mathtt{lu}^{-1}\mathtt{l}^{-1}\mathtt{u}^{-2}\mathtt{lfuf}^{-1}\mathtt{u}^{-1}\mathtt{l}^{-1}\mathtt{fru}^{-1}$
$\mathtt{r}^{-1}\mathtt{f}^{-1}\mathtt{l}^{-1}\mathtt{ul}$

8 Some Impossible Configurations

Following the examples in the previous section one might raise questions such as whether it is possible to flip a single edge cubie or to rotate a single corner cubie. We can use `GAP` to answer such questions.

8.1 Flip A Single Edge Cubie

```
gap> (4,10) in G;
false
```

8.2 Rotate A Single Corner Cubie

```
gap> (1,9,35) in G;
false
```

8.3 Swap Two Opposite Edge Cubies on the Same Face

```
gap> (4,5)(10,26) in G;
false
```

8.4 Swap Two Diametrically Opposite Edge Cubies

```
gap> (4,31)(10,45) in G;
false
```

8.5 Swap Two Opposite Corner Cubies on the Same Face

```
gap> (1,8)(9,25)(35,19) in G;
false
```

8.6 Swap Two Diametrically Opposite Corner Cubies

```
gap> (1,43)(9,24)(35,30) in G;
false
```

References

[1] The Rubik's Cube `https://www.rubiks.com`

[2] Online Rubik's Cube Simulator `https://rubiks-cu.be`

[3] GAP `https://www.gap-system.org`

[4] Analyzing Rubik's Cube with GAP `https://www.gap-system.org/Doc/Examples/rubik.html`