RUNNING SAUCY

Both saucy and shatter support the "--help" option for thorough descriptions of the available arguments to each program.

The general usage is "saucy graphfile". Statistics can be output to standard error via the "-s" flag.

Saucy does not support multigraphs (graphs with duplicate edges) nor DIMACS instances that produce multigraphs (CNF instances with duplicate literals within a clause or, in some cases, duplicate clauses).

INPUT FORMAT

Saucy uses a simple graph input format. Here is an example:

```
5 5 1
0 1
1 2
2 3
3 4
4 0
```

This graph represents a 5 vertex circle with all vertices given the same color; that is, the initial partition of vertices is unit.

Whitespace is ignored in the input; newlines can be present anywhere.

```
#vertices
#edges
#colors
[#colors-1 split locations]
[#edges vertex pairs]
```

Vertices are counted from 0.

Here is another example explaining the "split locations" component:

```
7 7 2 3
0 1
1 2
2 0
3 4
```

4 5
5 6
6 3

This represents the graph consisting of a triangle and a square. The vertices in the triangle {0,1,2} are colored differently than the vertices in the square {3,4,5,6}, since there are 2 colors, with a split at vertex 3.

This is admittedly an awkward way of specifying partitions; the C API is less awkward.

OUTPUT FORMAT

Symmetries are output one per line, as a product of nontrivial orbits. For the triangle/square graph:

(3 5)
(3 6)(4 5)
(0 1)
(0 2)

Each orbit is parenthesized, and the vertices are separated by spaces.

STATISTICS

When run with the -s flag, saucy outputs a number of statistics. Here is an example run with the 7pipe benchmark:

input file = ../graphs/dac/7pipe
vertices = 100668
edges = 1498971
group size = 3.677259e1158
levels = 474
nodes = 1889
generators = 473
total support = 119202
average support = 252.01
nodes per generator = 3.99
bad nodes = 0
cpu time (s) = 0.22

group size: the total number of symmetries of the graph. For highly symmetric graphs this number can be astronomical. Of course saucy does

not enumerate this entire set.

levels: the number of recursive decompositions performed by saucy before it reached a discrete partition. Indicates the maximum search tree depth.

nodes: number of executions of the partition refinement algorithm. For saerch nodes, we count refinement twice, since there are two partitions being manipulated in parallel.

generators: the size of the generating set computed by saucy.

total support: the support of a generator is the number of vertices it permutes. The total support is the sum of the supports of all generators.

average support: average number of vertices permuted by a generator. Indicates the sparsity of the generators found.

nodes per generator: average number of refinements required to find each generator. Indicates the success of this version of saucy over previous approaches, whose average would approximate the number of levels.

bad nodes: the number of times the search phase had to backtrack because of some failed mapping of vertices. Graphs with large amounts of bad nodes tend to be highly regular graphs, such that degree-based partition refinement is unable to distinguish vertices very well.

cpu time: the time taken by saucy, in seconds. This does not include the time required to read the graph into memory.