Nauty
A Brief Introduction

Christopher A. Wood

Department of Computer Science

Rochester Institute of Technology

`caw4567@cs.rit.edu`

RIT Folkman Group

March 10, 2013

Nauty
main functionality

A **graph canonical labeling** program with two main purposes

- determine the automorphism group of a vertex-colored graph (**nauty** = no automorphisms, yes?)
- generate non-isomorphic graphs

Graph automorphism

Graph automorphism: A vertex mapping that preserves connectivity between vertex and edge connectivity (e.g. an isomorphism onto itself)

The set of all automorphism on a graph is the *automorphism group*

[McKay]
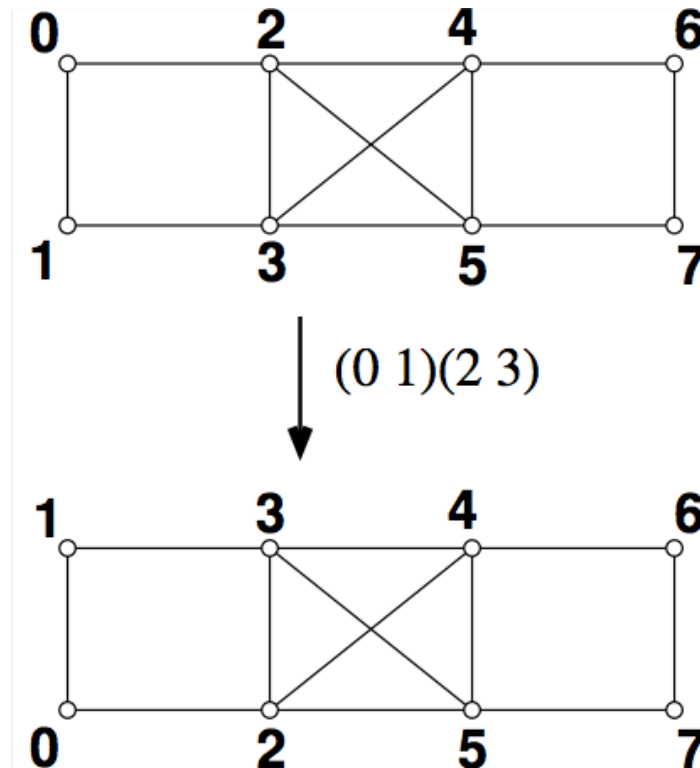
Graph isomorphism
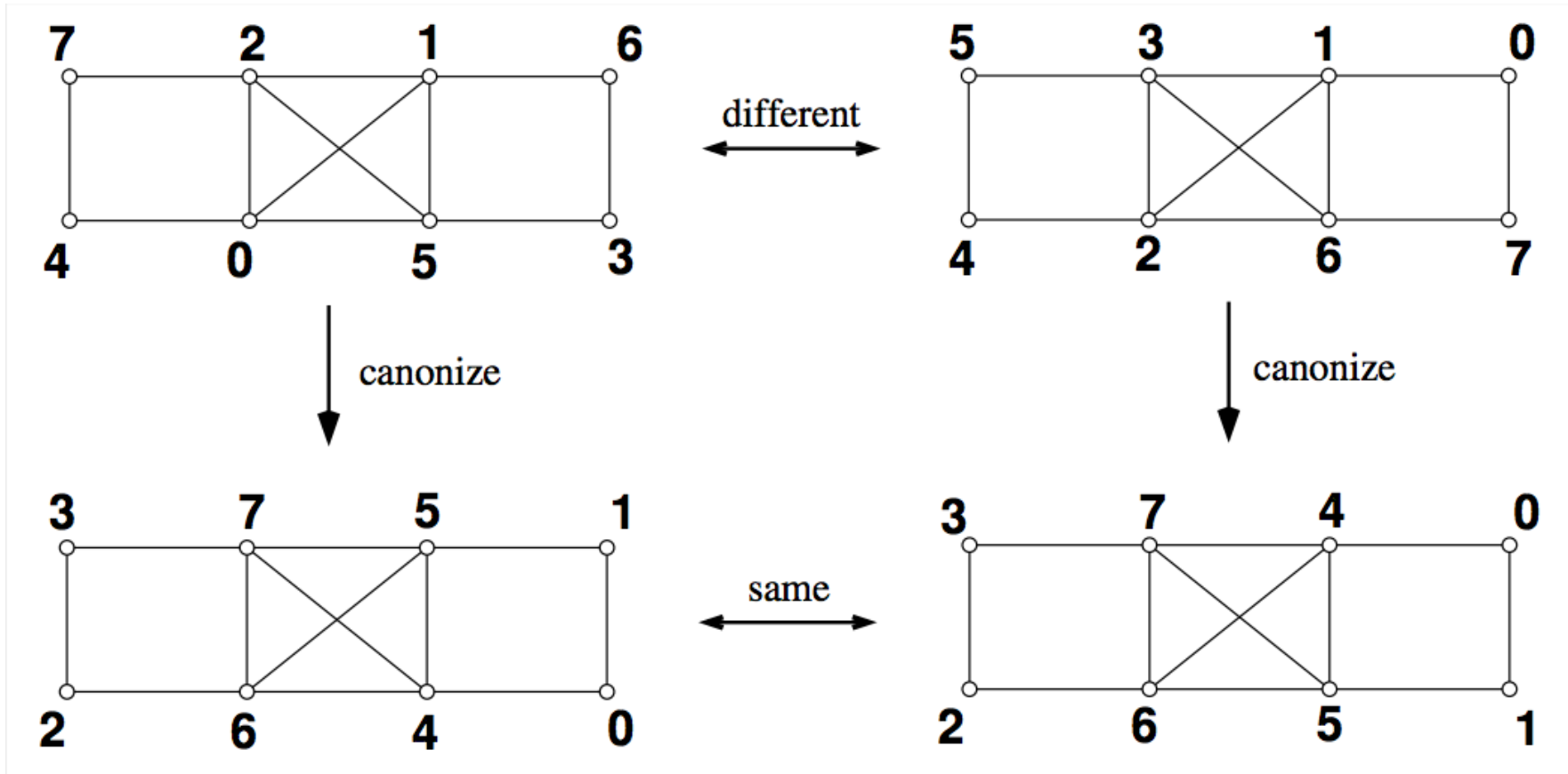definition

Graph isomorphism: A vertex mapping that $f : V(G) \to V(H)$ such that if $(u, v) \in E(G)$ then $(f(u), f(v)) \in E(H)$.

- The *canonical labeling* feature of nauty reduces all isomorphic graphs down to the same identical graph.
- This is how nonisomorphic graphs are found!

Graph isomorphism
finding isomorphic graphs through canonization

[McKay]

Using nauty
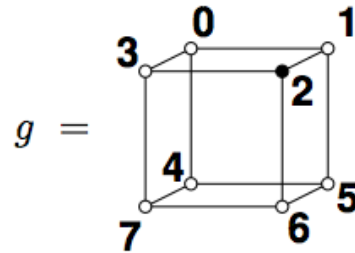getting started with dreadnaut

Nauty (and Traces) can be downloaded at:
`http://pallini.di.uniroma1.it`

Getting started (see what commands are available)...

```
raven-2:nauty caw$ dreadnaut
Dreadnaut version 2.4 (64 bits).
> h
```

Using nauty

a short example


```
> n=8 g
> 0: 1 3 4; enter the graph
  1: 2 5;
  2: 3 6;
  3: 7;
  4: 5 7;
  5: 6;
  6: 7.
> f=2 x
[fixing partition]
```

[McKay]

Scaling up
size limits

Nauty is limited by the word size n of the machine on which it is compiled (unfortunately)

- $n < 32$ `int` $\rightarrow 2^{15} - 3$ limit
- $n \leq 32$ `int` $\rightarrow 2^{30}$ limit

These are limits on the number of vertices (the order of the graph)

Wrapping nauty
writing our own code

We can easily use nauty procedures internally:

1. Include `nauty.h` and link to `nauty.c`, `nautil.c`, `naugraph.c`, `schreier.c`, and `naurng.c`.

2. Use any of the functions and macros defined in `nauty.h`!

```
#define MAXN 1000
#define MAXM 1000
#include "nauty.h"
int main(int argc, char *argv[]) {
    graph g[MAXN*MAXM];
    // your code goes here...
```

[McKay]

Useful utilities
automatic graph manipulation without writing your own code

- **geng** : generate small graphs
- **genbg** : generate small bicoloured graphs
- **gentourng** : generate small tournaments
- **directg** : generate small digraphs with given underlying graph
- **watercluster2** : a faster alternative to direct written by Gunnar Brinkmann
- **multig** : generate small multigraphs with given underlying graph
- **genrang** : generate random graphs
- **copyg** : convert format and select subset
- **labelg** : canonically label graphs

What else?

Consult the manual! :-)

References

- McKay, Brendan D., and Adolfo Piperno. *nauty and Traces User's Guide (Version 2.5)*. 2013.