

Computational Group Theory



Steve Linton

Centre for Interdisciplinary Research in
Computational Algebra

University of St Andrews

Groups

- The mathematician's handle on symmetry
- Key objects in mathematics
- Applications in physics, chemistry, comp. sci.,...
- One of the first areas of pure mathematics to be computerised
- here (almost always) finitely generated and discrete



“There will be positively no internal alterations to be made even if we wish suddenly to switch from calculating the energy levels of the neon atom to the enumeration of groups of order 720.”

Alan Turing (1945)

Algorithms – History



- Todd & Coxeter, 1936: “A Practical Method for Enumerating Cosets of a Finite Abstract Group”
 - formalised a type of computation going back to 1900
- 1950s – First digital computer experiments
- 1970s – First software systems: Aachen-Sydney Group System, Cayley, CAS, SOGOS, SPAS, Meataxe
 - Very many algorithms – permutation groups, p-groups, fp groups, repn theory
- Since then – increasing sophistication in software and algorithms
 - Mature integrated software: GAP and MAGMA

Examples of the State of the Art



- We can compute freely with
 - Permutation groups on up to a million or so points
 - Matrix groups over finite fields
 - Smallish rational matrix groups
 - Crystallographic groups
- This means we can compute with elements, subgroups and homomorphisms:
 - centre, centralizer, normalizer, composition series, Sylow subgroups, conjugacy classes, image and kernel of mappings, automorphism group, coset reps, ...

Not Just Groups

- Modern group theoretic computations often need other things
 - polynomial arithmetic, Gröbner bases, etc.
 - Number theory – factorisation, discrete log,
 - algebras – Lie algebras for p-groups, associative algebras for representation theory
- Also the approaches and tools developed for group theory can be applied elsewhere
 - nearrings, semigroups and monoids
 - Lie algebras for their own sake or for applications

Software

GAP

www.gap-system.org

- Aachen, Braunschweig, Fort Collins, St Andrews
- Free (GPL)
- C kernel, most of system in GAP language
- 60+ contributed extension packages
 - Including links to specialised stand-alones

MAGMA

<http://magma.maths.usyd.edu.au/magma/>

- Sydney
- Charge for costs of distribution and support
- Mainly C system
- Increasing number of Magma language packages

GAP in Action

```
gap> AvgOrder :=
> g->Sum(ConjugacyClasses(g),
> c-> Size(c)*Order(Representative(c)))/
> Size(g);
function( g ) ... end
gap> AvgOrder(MathieuGroup(11));
53131/7920
gap> ForAny(AllSmallGroups([2..100]),
> g->IsInt(AvgOrder(g)));
false
```

- Would look similar in Magma
- Text based “read eval print” UI
- Program with natural mathematical concepts like “Size”, “Conjugacy Classes”
 - Multiple algorithms behind the scenes
- Databases
 - in this case “Small groups”
 - Many others exist

CGT for Applications

- Symmetry is a consideration in many computational problems
 - Combinatorial search and optimisation to avoid exploring “the same” solutions many time
 - Symmetries of crystals and molecules in computational physics or chemistry
 - Representation theory arises in particle and solid state physics.
- We have algorithms and implementations which can answer a lot of questions of these kinds
 - Would be interesting to see if they help with scientific software applications
 - Maybe find new mathematical questions.

Linking Software

- Neither GAP nor Magma is a subroutine library
 - Designed to be used as interactive stand-alone software
 - This has been a big obstacle to incorporating these techniques in other applications
- Now possible to set up GAP or Magma based servers (locally or on the internet) and connect using
 - Standard (SOAP/http) Web services
 - Or directly via a simple SCSCP protocol (C/Java client libraries exist).
 - OK if the computations are reasonably coarse-grained.
- Neither GAP nor Magma is parallelised (yet).



Thanks for listening.