Computational group theory for young group theorists

Jack Schmidt

University of Kentucky

2008-04-16

Computational group theory is a wonderful branch of science studying how to ask questions in group theory in ways amenable to computation and the corresponding methods of answering them algorithmically. Many of the results of this field are made available in the computer software GAP.

Outline

- What is computational group theory?
- How do I ask questions in group theory amenable to computation?
- How do I get GAP to answer them?

Origin of group theory

- Group theory began as an applied science
- Applied to field theory by Jordan, Ruffini, Abel, Galois, Sathaye
- Applied to topology and geometry by Dehn, Todd, Coxeter, Nielsen, Anton
- Applied to partial differential equations by Lie, Cartan, Burnside, Hislop
- Formalism of abstract group not discussed until all these applications had firmly established groups as applied science
- Applied fields are nice, because people actually expect answers

Computational group theory

- Early contributors were Dehn, Schreier, Todd, Coxeter (1910-1940)
- They developed the necessary questions and answers for one type of group: finitely presented groups
- Many computations done by others for permutation groups, but ad hoc
- The founding fathers of CGT published the foundations in the 1960s: Sims, Neubüser
- Sims's strong generating set and base for permutation groups
- 1980s saw great improvements to finite solvable groups
- 2000s see great improvements to matrix groups

How do you ask the questions?

- How do you tell somebody:
  1. which group you are talking about?
  2. which element of that group you are talking about?
  3. which homomorphism you are talking about?
- Consider the problem for integers:
  1. **Tally marks**: commonly known integer 1, and a very simple rule for producing all others
  2. **Roman numerals**: commonly known integers I, V, X, L, C, M, and rules for producing the other integers from them
  3. **Radix notation**: specify a_i to denote $\sum a_i b^i$ where b is the chosen base and $0 \leq a_i < b$ is a thus a "commonly known" integer.
  4. **Prime factorization**: commonly known integers the primes, and then "smaller" integers used for exponents

Are these amenable to computation?

- Tally marks:
  - \+ wonderfully precise
  - \+ addition algorithm exists and is linear in input length
  - \+ multiplication algorithm exists and is quadratic in input length
  - – "big integers" (by today's standard) are hard to represent

---

- Roman numerals:
  - – Multiple representations for same number
  - – Addition algorithm complicated and slow
  - – Multiplication algorithm complicated and slow
  - – "big integers" (by today's standard) are hard to represent

Are these amenable to computation?

- Radix notation:
  - \+ Precise for integers
  - \+ addition algorithm exists and is linear in input length
  - \+ multiplication algorithm exists and is quadratic in input length
  - \+ "big integers" (by today's standard) are easy to represent

---

- Prime power factorization:
  - \+ Precise for integers
  - − addition algorithm exists, but is **exponential** in input length
  - \+ multiplication algorithm exists, and is **linear** in input length
  - − Some "big integers" (by today's standard) are hard to represent

How to specify the group:

- Have to rely on commonly known groups, and construct from there
- Which commonly known groups and which constructions?
- A standard set of commonly known groups are:
  1. the **general linear group**,
  2. the **symmetric group**,
  3. and the **free group**
- All other groups are specified as **quotients** of **subgroups** of these
- Notations for elements are easy to agree on

Which subgroup? Which quotient?

- Subgroup specified by:
  1. finite list of **generators** (most common),
  2. or finite set of **equations** (stabilizer, algebraic group)
- Quotient specified by:
  1. finite list of **generators** as normal subgroup (most common),
  2. or finite set of **equations** (kernel)

Examples:

- The dihedral group can be represented many ways:
- The subgroup of Sym(4) generated by $\{(1,2,3,4),(2,4)\}$
- The subgroup of $\mathrm{GL}(2,\mathbb{Z})$ which leaves the standard bilinear form invariant
- The quotient group of the free group on $\{x,y\}$ by the normal subgroup generated by $\{x^4, y^2, (xy)^2\}$
- The second example in detail: $\{A \in \mathrm{GL}(2,\mathbb{Z}) : A \cdot A^{\mathrm{T}} = 1\}$

$$\left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} : a,b,c,d \in \mathbb{Z} | \, (ad-bc)^2 = 1, \right.$$

$$aa + bb = 1,$$

$$ac - bd = 0,$$

$$\left. cc + dd = 1 \right\}$$

How to ask GAP questions about permutations?

- Permutation are specified in cycle notation on the set of positive integers

```
gap> pi := (1,2,3)(4,5);
```

- Permutations π act on positive integers n as n^π

```
gap> 1^pi;
2
```

- Permutations multiply so that $(n^\pi)^\sigma = n^{\pi \cdot \sigma}$

```
gap> sigma := (5,6);; pi*sigma;
(1,2,3)(4,6,5)
```

How to ask GAP questions about permutation groups?

- Permutation groups are specified by generators:

```
gap> myd8 := Group( (1,2,3,4), (2,4) );
```

- Can find the order of the group:

```
gap> Order(myd8);
8
```

- May be described in text:

```
gap> StructureDescription(myd8);
"D8"
```

- Can find centers, derived subgroup:

```
gap> Center(myd8);
Group([ (1,3)(2,4) ])
gap> DerivedSubgroup(myd8);
Group([ (1,3)(2,4) ])
```

More questions and answers

- Can find Sylow subgroups:

```
gap> p := SylowSubgroup(SymmetricGroup(4),2);
Group([ (1,2), (3,4), (1,3)(2,4) ])
gap> StructureDescription(p);
"D8"
```

- Can check equality:

```
gap> p = myd8;
false
```

- Can check isomorphism:

```
gap> IsomorphismGroups(p,myd8);
[ (1,2), (3,4), (1,3)(2,4) ] -> [ (2,4), (1,3),
(1,4)(2,3) ]
```

More questions and answers

- Can list all of the elements:

```
gap> Elements(myd8);
[ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3),
(1,3)(2,4), (1,4,3,2), (1,4)(2,3) ]
gap> Elements(p);
[ (), (3,4), (1,2), (1,2)(3,4), (1,3)(2,4),
(1,3,2,4), (1,4,2,3), (1,4)(2,3) ]
```

- Can check if the group is transitive:

```
gap> IsTransitive(myd8);
true
gap> IsPrimitive(myd8);
false
gap> Stabilizer(myd8,1);
Group([ (2,4) ])
```

More questions and answers

- Can check if the group is abelian:

```
gap> IsAbelian(myd8);
false
gap> IsNilpotent(myd8);
true
gap> IsSolvable(g8);
true
```

- Can find its lower central series:

```
gap> LowerCentralSeries(myd8);
[ D8, Group([ (1,3)(2,4) ]), Group(()) ]
gap> UpperCentralSeries(myd8);
[ Group([ (1,3)(2,4), (2,4), (1,2,3,4) ]), Group([
(1,3)(2,4) ]), Group(()) ]
```

Exercises on groups

- Is $(1, 2, 3)$ in myd8?
- What sort of group is generated by $\{(1, 2)(3, 4)(5, 6)(7, 8), (1, 5)(2, 6)(3, 7)(4, 8)\}$?
- Is $(1, 2, 3)$ in the group generated by $\{(1, 2, 3, 4, 5), (2, 4, 5)\}$?
- Is this group simple? supersoluble?
- Which alternating groups have subgroups of order 6?

Summary

- Group theory questions can have answers!
- You can learn to ask questions that are amenable to computation!
- There is even software to answer such questions!

The End