Using Sage to Teach Group Theory

Rob Beezer
beezer@ups.edu

Department of Mathematics and Computer Science
University of Puget Sound

Sage Edu Days 1
Clay Mathematics Institute
December 5, 2009

Undergraduate Group Theory

- Fall: Group Theory
- Spring: Rings, Modules, Fields, Galois Theory
- Nine students, all seniors
  - 4 Mathematics majors
  - 3 Math, Computer Science double majors
  - 1 Physics major
  - 1 English major
- Introductory programming required for a math major
- Abstract Algebra: Theory and Applications, by Tom Judson
  - Open-source, at `abstract.pugetsound.edu`
  - Similar to Gallian
  - Topics similar to Herstein
  - One chapter a week, one Sage exercise set each week

Why Sage?

- Simpler syntax than GAP, almost as powerful for our purpose
- Will be even more valuable for ring theory, field extension topics
- Notebook interface
    - Easy setup (at minimum, `sagenb.org`)
    - Easy for students to learn
    - Students can comment on their work with LaTeX
- Python language for simple programming
- Skills transfer
    - to other courses
    - to other areas of mathematics
    - to other areas of life

Goals

- Learn basics of
  - Sage
  - Python
  - LaTeX
- Become comfortable with calculations in groups
- Form conjectures from long sequences of computations
- Experiment!
- Understand constructive proofs

Chapter 0

- Judson: Proofs, Sets, Equivalence relations
- One class session on Sage and notebook
  - Getting started
  - Saving, emailing a worksheet
  - Help: tab-completion, manuals
- Exercise: Email me one interesting computation
- Developer Project: Implement equivalence relations (wrap GAP?)

Chapter 1

- Judson: Induction, Divisibility, Primes, GCD, etc
- Exercise
  - Build, check primes
  - Compute GCDs
  - Relatively prime pair as linear combination equaling 1
  - Factorization
  - Divisibility checks (mod, div operations)
- Instructions walked them through these steps

Chapter 2

- Judson: Integers mod n, Symmetries of plane figures
- Judson: Basics of groups, subgroups
- Exercise
  - Permutation group constructions, `G = SymmetricGroup(3)`
  - Properties: `G.order()`, `G.abelian()`
  - Cayley tables: `G.cayley_table()`
  - Formatted discussion of Cayley tables, S_3 vs. C_6
  - Optional: create a subgroup using an abundance of generators using `H = G.subgroup( [gens] )`
    Check with `G.is_subgroup(H)`
- Still learning basics, asked for discussion, some experimentation
- Developer Project: Improve Cayley tables

Chapter 3

- Judson: Cyclic groups, their subgroup structure
- Exercise
  - Group of units under multiplication mod 40, mod 49, mod 35 (Only mod 49 is cyclic)
  - Ring: `R = Integers(40)`
  - Group: `U = R.list_of_elements_of_multiplicative_group()`
  - Explore orders, cyclic-ness with Python loops
  - Open-ended question: conjecture about structure of $U(n)$
- Basic Python loops, more discussion, more speculation
- Student: Show a group is non-cyclic by lack of generators **OR** two elements of order two
- Developer Project: Implement the group of units mod n on top of abelian groups

Chapter 4

- Judson: Permutation groups, cycle structure, alternating group
- Exercise
  - Group and element constructions, using cycle notation
  - Basic computations: powers, inverses, signs, orders
  - Cyclic subgroups via single generator
  - Experiment: Build all subgroups of A_4 "manually"
  - Crash Sage?
    List PermutationGroup([“(1,3)(4,5)”, “(1,3)(2,5,8)(4,6,7,9,10)”])
    (order 80 640)
- Computational proficiency in permutation groups
- Brute-force experimentation to find subgroups

Chapter 5

- Judson: Cosets, Lagrange's Theorem
- Exercise
  - Provided a routine to make all subgroups
  - Experiment: find group, divisor of subgroup such that there is no subgroup of that order (and not A_4 example from class)
  - Other items not worth talking about
- Developer Project: all subgroups of a group (GAP gives representatives of conjugacy classes of subgroups)

Chapter 8

- Judson: Isomorphisms
- Exercise
  - Permutation representations via Cayley's Theorem (the left or right regular representation)
  - Practice: Quaternions
  - $Z_2 \times Z_4$
  - Units mod 24
  - Use Sage commands to verify representation
- Suggested pencil and paper, with Sage as calculator
- Students: Easiest for students who wrote general code
- Students: test question was easier

Chapter 9a

- Judson: Normal subgroups, quotient groups
- Exercise
  - For A_4 and D_4 (dihedral group of order 8):
  - All subgroups with provided routine
  - Then test left and right coset equality (brute-force)
  - Check with `G.normal_subgroups()`
  - All quotient groups in A_4 (`A4.quotient_group(N)`)
- Doubly-nested loops, sorted lists, logic for equalities
- Developer Project: Quotient groups with cosets as elements, not isomorphic permutation groups

Chapter 9b

- Judson: Homomorphisms, isomorphism theorems, composition series
- Exercise
  - More normal subgroups, plus simple groups
  - All normal subgroups: Z_{40} versus Z_{41}
  - All normal subgroups of D_n (dihedral order $2n$)
    Conjecture pattern (one per divisor of n (almost!), parity discrepancy
  - Higman-Sims group: two generators in S_{100} at Atlas web page
    Order is 44 352 000, Sage gets simplicity quickly
- Formulate conjectures
- Work with large examples

Chapter 11

- Judson: Classify finite abelian groups
- Exercise
  - Walk through constructive proof
  - Uses elements of maximal order to build internal direct products
  - Units mod $441 = 3^2 7^2 \quad (Z_2 \times Z_2 \times Z_3 \times Z_3 \times Z_7)$
  - Units mod $2312 = 2^3 17^2 \quad (Z_{16} \times Z_4 \times Z_{17})$
  - 241 in units mod 441 as product of powers of generators
    brute-force a discrete-log type computation (5 generators)
- Required students read, understand proof
- Student: Very automated approach,
  and then found a very significant typo in book's proof

Chapter 12

- Judson: Group actions
- Class: Half-day on automorphism groups of graph
- Exercise
  - Automorphisms of 4-D cube, and "symmetric" graph on 8 vertices
    `A = G.automorphism_group()`
  - Orbits: `A.orbits()`, see one vertex-transitive, one not
  - Recognize "different" vertices in smaller graph
  - Stabilizers: `B = A.stabilizer(1)`
  - Orbits of stabilizers: `B.orbits()`
  - Orbits of stabilizers are refinement of distance partition
    (equal for 4-D cube)
  - Higman-Sims group is transitive (see this with orbits)
- More conjectures, experimentation
- Lots of good graph theory in Sage, e.g. constructors, graphics
- Student: having isomorphic stabilizers is an equivalence relation?

Chapter 13

- Judson: Sylow theorems
- Exercise
  - Sage/GAP gives **one** Sylow p-subgroup per prime
  - Conjugate to build them all, remove duplicates (Second Sylow Theorem)
  - For A_5 and dihedral group D_{36} confirm conditions on number of Sylow subgroups (Third Sylow Theorem)
- Uses now-familiar programming techniques
- Employ a theorem, confirm a theorem

Student Worksheet

Student Worksheet at `http://sagenb.org`

Conclusions

- Better performance on certain test questions, especially relating to constructive proofs (regular representations, finite abelian groups)
- Programming was not an impediment
- Some nontrivial computations (even massive)
- Students: Some good, thoughtful conjectures
- Students: Some interesting algorithmic approaches
- Developer Project: even better tools for managing homework
- Most returning for more in the spring — I'm looking forward to it

CREDITS

David Joyner, Robert Bradshaw, William Stein, Robert Miller, Tom Judson

This talk

buzzard.ups.edu/talks.html

Exercises

buzzard.ups.edu/courses/2009fall/m433-sage-exercises.pdf

Group Theory and Sage Primer

abstract.ups.edu/sage-aata.html