

DISCRETE MATHEMATICS AND ITS APPLICATIONS
Series Editor KENNETH H. ROSEN

HANDBOOK OF COMPUTATIONAL GROUP THEORY

DEREK F. HOLT

BETTINA EICK

EAMONN A. O'BRIEN



CHAPMAN & HALL/CRC

A CRC Press Company

Boca Raton London New York Washington, D.C.

Contents

Notation and displayed procedures	xvi
1 A Historical Review of Computational Group Theory	1
2 Background Material	9
2.1 Fundamentals	9
2.1.1 Definitions	9
2.1.2 Subgroups	11
2.1.3 Cyclic and dihedral groups	12
2.1.4 Generators	13
2.1.5 Examples — permutation groups and matrix groups	13
2.1.6 Normal subgroups and quotient groups	14
2.1.7 Homomorphisms and the isomorphism theorems	15
2.2 Group actions	17
2.2.1 Definition and examples	17
2.2.2 Orbits and stabilizers	19
2.2.3 Conjugacy, normalizers, and centralizers	20
2.2.4 Sylow's theorems	21
2.2.5 Transitivity and primitivity	22
2.3 Series	26
2.3.1 Simple and characteristically simple groups	26
2.3.2 Series	27
2.3.3 The derived series and solvable groups	27
2.3.4 Central series and nilpotent groups	29
2.3.5 The socle of a finite group	31
2.3.6 The Frattini subgroup of a group	32
2.4 Presentations of groups	33
2.4.1 Free groups	33
2.4.2 Group presentations	36
2.4.3 Presentations of group extensions	38
2.4.4 Tietze transformations	40
2.5 Presentations of subgroups	41
2.5.1 Subgroup presentations on Schreier generators	41
2.5.2 Subgroup presentations on a general generating set	44
2.6 Abelian group presentations	46

2.7	Representation theory, modules, extensions, derivations, and complements	48
2.7.1	The terminology of representation theory	49
2.7.2	Semidirect products, complements, derivations, and first cohomology groups	50
2.7.3	Extensions of modules and the second cohomology group	52
2.7.4	The actions of automorphisms on cohomology groups	54
2.8	Field theory	56
2.8.1	Field extensions and splitting fields	56
2.8.2	Finite fields	58
2.8.3	Conway polynomials	59
3	Representing Groups on a Computer	61
3.1	Representing groups on computers	61
3.1.1	The fundamental representation types	61
3.1.2	Computational situations	62
3.1.3	Straight-line programs	64
3.1.4	Black-box groups	65
3.2	The use of random methods in CGT	67
3.2.1	Randomized algorithms	67
3.2.2	Finding random elements of groups	69
3.3	Some structural calculations	72
3.3.1	Powers and orders of elements	72
3.3.2	Normal closure	73
3.3.3	The commutator subgroup, derived series, and lower central series	73
3.4	Computing with homomorphisms	74
3.4.1	Defining and verifying group homomorphisms	74
3.4.2	Desirable facilities	75
4	Computation in Finite Permutation Groups	77
4.1	The calculation of orbits and stabilizers	77
4.1.1	Schreier vectors	79
4.2	Testing for $\text{Alt}(\Omega)$ and $\text{Sym}(\Omega)$	81
4.3	Finding block systems	82
4.3.1	Introduction	82
4.3.2	The Atkinson algorithm	83
4.3.3	Implementation of the class merging process	85
4.4	Bases and strong generating sets	87
4.4.1	Definitions	87
4.4.2	The Schreier-Sims algorithm	90
4.4.3	Complexity and implementation issues	93
4.4.4	Modifying the strong generating set — shallow Schreier trees	95

4.4.5	The random Schreier-Sims method	97
4.4.6	The solvable BSGS algorithm	98
4.4.7	Change of base	102
4.5	Homomorphisms from permutation groups	105
4.5.1	The induced action on a union of orbits	105
4.5.2	The induced action on a block system	106
4.5.3	Homomorphisms between permutation groups	107
4.6	Backtrack searches	108
4.6.1	Searching through the elements of a group	110
4.6.2	Pruning the tree	113
4.6.3	Searching for subgroups and coset representatives	114
4.6.4	Automorphism groups of combinatorial structures and partitions	118
4.6.5	Normalizers and centralizers	121
4.6.6	Intersections of subgroups	124
4.6.7	Transversals and actions on cosets	126
4.6.8	Finding double coset representatives	131
4.7	Sylow subgroups, p -cores, and the solvable radical	132
4.7.1	Reductions involving intransitivity and imprimitivity	133
4.7.2	Computing Sylow subgroups	134
4.7.3	A result on quotient groups of permutation groups	137
4.7.4	Computing the p -core	138
4.7.5	Computing the solvable radical	140
4.7.6	Nonabelian regular normal subgroups	141
4.8	Applications	143
4.8.1	Card shuffling	144
4.8.2	Graphs, block designs, and error-correcting codes	145
4.8.3	Diameters of Cayley graphs	147
4.8.4	Processor interconnection networks	148
5	Coset Enumeration	149
5.1	The basic procedure	150
5.1.1	Coset tables and their properties	151
5.1.2	Defining and scanning	152
5.1.3	Coincidences	156
5.2	Strategies for coset enumeration	162
5.2.1	The relator-based method	162
5.2.2	The coset table-based method	165
5.2.3	Compression and standardization	167
5.2.4	Recent developments and examples	168
5.2.5	Implementation issues	170
5.2.6	The use of coset enumeration in practice	171
5.3	Presentations of subgroups	173
5.3.1	Computing a presentation on Schreier generators	173
5.3.2	Computing a presentation on the user generators	178

5.3.3	Simplifying presentations	184
5.4	Finding all subgroups up to a given index	188
5.4.1	Coset tables for a group presentation	189
5.4.2	Details of the procedure	190
5.4.3	Variations and improvements	196
5.5	Applications	198
6	Presentations of Given Groups	199
6.1	Finding a presentation of a given group	199
6.2	Finding a presentation on a set of strong generators	205
6.2.1	The known BSGS case	205
6.2.2	The Todd-Coxeter-Schreier-Sims algorithm	207
6.3	The Sims ‘Verify’ algorithm	208
6.3.1	The single-generator case	209
6.3.2	The general case	213
6.3.3	Examples	217
7	Representation Theory, Cohomology, and Characters	219
7.1	Computation in finite fields	220
7.2	Elementary computational linear algebra	221
7.3	Factorizing polynomials over finite fields	226
7.3.1	Reduction to the squarefree case	228
7.3.2	Reduction to constant-degree irreducibles	229
7.3.3	The constant-degree case	229
7.4	Testing KG -modules for irreducibility — the Meataxe	230
7.4.1	The Meataxe algorithm	230
7.4.2	Proof of correctness	234
7.4.3	The Ivanyos-Lux extension	235
7.4.4	Actions on submodules and quotient modules	235
7.4.5	Applications	236
7.5	Related computations	237
7.5.1	Testing modules for absolute irreducibility	237
7.5.2	Finding module homomorphisms	241
7.5.3	Testing irreducible modules for isomorphism	244
7.5.4	Application — invariant bilinear forms	245
7.5.5	Finding all irreducible representations over a finite field	246
7.6	Cohomology	248
7.6.1	Computing first cohomology groups	249
7.6.2	Deciding whether an extension splits	253
7.6.3	Computing second cohomology groups	254
7.7	Computing character tables	255
7.7.1	The basic method	256
7.7.2	Working modulo a prime	257
7.7.3	Further improvements	260

7.8	Structural investigation of matrix groups	264
7.8.1	Methods based on bases and strong generating sets	264
7.8.2	Computing in large-degree matrix groups	268
8	Computation with Polycyclic Groups	273
8.1	Polycyclic presentations	274
8.1.1	Polycyclic sequences	274
8.1.2	Polycyclic presentations and consistency	278
8.1.3	The collection algorithm	280
8.1.4	Changing the presentation	284
8.2	Examples of polycyclic groups	286
8.2.1	Abelian, nilpotent, and supersolvable groups	286
8.2.2	Infinite polycyclic groups and number fields	288
8.2.3	Application — crystallographic groups	289
8.3	Subgroups and membership testing	290
8.3.1	Induced polycyclic sequences	291
8.3.2	Canonical polycyclic sequences	296
8.4	Factor groups and homomorphisms	298
8.4.1	Factor groups	298
8.4.2	Homomorphisms	299
8.5	Subgroup series	300
8.6	Orbit-stabilizer methods	302
8.7	Complements and extensions	304
8.7.1	Complements and the first cohomology group	304
8.7.2	Extensions and the second cohomology group	307
8.8	Intersections, centralizers, and normalizers	311
8.8.1	Intersections	311
8.8.2	Centralizers	313
8.8.3	Normalizers	314
8.8.4	Conjugacy problems and conjugacy classes	316
8.9	Automorphism groups	317
8.10	The structure of finite solvable groups	320
8.10.1	Sylow and Hall subgroups	320
8.10.2	Maximal subgroups	322
9	Computing Quotients of Finitely Presented Groups	325
9.1	Finite quotients and automorphism groups of finite groups	326
9.1.1	Description of the algorithm	326
9.1.2	Performance issues	332
9.1.3	Automorphism groups of finite groups	333
9.2	Abelian quotients	335
9.2.1	The linear algebra of a free abelian group	335
9.2.2	Elementary row operations	336
9.2.3	The Hermite normal form	337

9.2.4	Elementary column matrices and the Smith normal form	341
9.3	Practical computation of the HNF and SNF	347
9.3.1	Modular techniques	347
9.3.2	The use of norms and row reduction techniques	349
9.3.3	Applications	352
9.4	p -quotients of finitely presented groups	353
9.4.1	Power-conjugate presentations	353
9.4.2	The p -quotient algorithm	355
9.4.3	Other quotient algorithms	364
9.4.4	Generating descriptions of p -groups	364
9.4.5	Testing finite p -groups for isomorphism	371
9.4.6	Automorphism groups of finite p -groups	371
9.4.7	Applications	372
10	Advanced Computations in Finite Groups	375
10.1	Some useful subgroups	376
10.1.1	Definition of the subgroups	376
10.1.2	Computing the subgroups — initial reductions	377
10.1.3	The O’Nan-Scott theorem	378
10.1.4	Finding the socle factors — the primitive case	379
10.2	Computing composition and chief series	381
10.2.1	Refining abelian sections	381
10.2.2	Identifying the composition factors	382
10.3	Applications of the solvable radical method	383
10.4	Computing the subgroups of a finite group	385
10.4.1	Identifying the TF-factor	386
10.4.2	Lifting subgroups to the next layer	387
10.5	Application — enumerating finite unlabelled structures	390
11	Libraries and Databases	393
11.1	Primitive permutation groups	394
11.1.1	Affine primitive permutation groups	395
11.1.2	Nonaffine primitive permutation groups	396
11.2	Transitive permutation groups	397
11.2.1	Summary of the method	397
11.2.2	Applications	399
11.3	Perfect groups	400
11.4	The small groups library	402
11.4.1	The Frattini extension method	404
11.4.2	A random isomorphism test	405
11.5	Crystallographic groups	407
11.6	The “ATLAS of Finite Groups”	409

12	Rewriting Systems and the Knuth-Bendix Completion Process	411
12.1	Monoid presentations	412
12.1.1	Monoids and semigroups	412
12.1.2	Free monoids and monoid presentations	415
12.2	Rewriting systems	417
12.3	Rewriting systems in monoids and groups	423
12.4	Rewriting systems for polycyclic groups	426
12.5	Verifying nilpotency	429
12.6	Applications	431
13	Finite State Automata and Automatic Groups	433
13.1	Finite state automata	434
13.1.1	Definitions and examples	434
13.1.2	Enumerating and counting the language of a dfa	437
13.1.3	The use of fsa in rewriting systems	439
13.1.4	Word-acceptors	441
13.1.5	2-variable fsa	442
13.1.6	Operations on finite state automata	442
13.1.6.1	Making an fsa deterministic	443
13.1.6.2	Minimizing an fsa	444
13.1.6.3	Testing for language equality	446
13.1.6.4	Negation, union, and intersection	447
13.1.6.5	Concatenation and star	447
13.1.7	Existential quantification	448
13.2	Automatic groups	451
13.2.1	Definitions, examples, and background	451
13.2.2	Word-differences and word-difference automata	453
13.3	The algorithm to compute the shortlex automatic structures	456
13.3.1	Step 1	457
13.3.2	Step 2 and word reduction	459
13.3.3	Step 3	460
13.3.4	Step 4	462
13.3.5	Step 5	464
13.3.6	Comments on the implementation and examples	466
13.4	Related algorithms	468
13.5	Applications	469
	References	471
	Index of Displayed Procedures	497
	Author Index	499
	Subject Index	503