Digital Collections of Examples in Mathematical Sciences

James Davenport[1]

University of Bath

23 June 2021

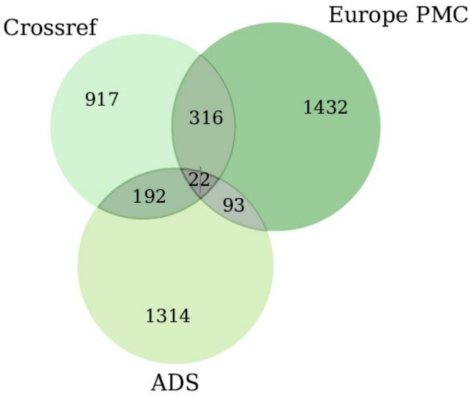
[1] Partially Supported by EPSRC Grant EP/T015713/1

Plan of Talk

1. Important Collections in Pure Mathematics
2. Important Test Suites in SAT/SMT
3. The Lack of Test Suites elsewhere
4. Way Forward?

Data Citation

- Is a mess in practice [vdSNI+19]: only 1.16% of dataset DOIs in Zenodo are cited (and 98.5% of these are self-citation).
- Is poorly harvested: [vdSNI+19, Figure 5].



so there are between 4,000–20,000 data sets waiting to be cited

Data Citation

- Is a mess in practice [vdSNI+19]: only 1.16% of dataset DOIs in Zenodo are cited (and 98.5% of these are self-citation).
- Is poorly harvested: [vdSNI+19, Figure 5].
- Is still a subject of some uncertainty: [MN12, KS14]
- Changes are still being proposed [DPS+20]
- *de facto* people cite a paper if they can find one.

Important Databases in Pure Mathematics

OEIS Online Encyclopedia of Integer Sequences [Slo03];

Long time at `http://www.research.att.com/~njas/sequences`; now at `https://oeis.org/`.

* Recommended citation: "N. J. A. Sloane, editor, The On-Line Encyclopedia of Integer Sequences, published electronically at `https://oeis.org`, [date]".

But you have to search the website to find it!

Group Theory (as an example)

- The Classification of Finite Simple Groups
- The Transitive Groups acting on n points: [BM83] ($n \leq 11$); [Roy87] ($n = 12$); [But93] ($n = 14, 15$); [Hul96] ($n = 16$); [Hul05] ($17 \leq n \leq 31$); [CH08] ($n = 32$).
- These are in GAP (and MAGMA), except that $n = 32$ isn't in the default build.

\+ These are a really great resource (if that's what you want)

– How do you cite them? "[The21, GAP `transgrp` library]"?

Also Other libraries such as primitive groups

Group Theory is "easy": for a given n there are a finite number and we "just" have to list them.

SAT Solving

SAT solving, normally seen as solving a Boolean expression written in CNF. Given a 3-literals/clause CNF satisfiability problem,

$$\underbrace{(l_{1,1} \vee l_{1,2} \vee l_{1,3})}_{\text{Clause 1}} \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge \cdots \wedge (l_{N,1} \vee l_{N,2} \vee l_{N,3}),$$

where $l_{i,j} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots\}$, is it satisfiable? In other words, is there an assignment of $\{T, F\}$ to the x_i such that all the clauses are *simultaneously* true.

3-SAT: the quintessential NP-complete problem [Coo66]. 2-SAT is polynomial, and k-SAT for $k > 3$ is polynomial-transformable into 3-SAT. In practice we deal with SAT — i.e. no limitations on the length of the clauses.

Let n be the number of i such that x_i (and/or $\overline{x_i}$) actually occur. Typically n is of a similar size to N.

SAT Solving

Despite being NP-complete, nearly all examples are easy (e.g. [KS00]),
either easily solved (SAT) or easily proved insoluble (UNSAT) and for random problems there seems to be a distinct phase transition between the two: [GW94, AP04, AP06].
This means that constructing difficult examples is itself difficult, and a research area in itself: [Spe15, BC18].
SAT solving has many applications, so we want effective solvers for "real" problems, not just "random" ones.
Fundamental question: what does this mean?

SAT Contests: http://www.satcompetition.org

Been run since 2002. In the early years, distinct tracks for Industrial/Handmade/Random problems: this has been abandoned.
The methodology is that the organisers accept submissions (from contestants and others), then produce a list of problems (in a standard format) and set a time (and memory) limit, and see how many of the problems the submitted systems can solve on the contest hardware.
SAT is easy to certify (just produce a list of values), UNSAT is much harder, but since 2013 the contest has required proofs of UNSAT for the UNSAT track, and since 2020 in all tracks, in DRAT: a specified format (some have been $>$ 100GB).
The general feeling is that these contests have really pushed the development of SAT solvers, roughly speaking $\times 2$/year. For comparison, Linear Programming has done $\times 1.8$ over a greater timeline [Bix15].

SMT: life beyond SAT

Consider a theory T, with variables y_j, and various Boolean-valued statements in T of the form $F_i(y_1, \ldots, y_n)$, and a CNF with $F_i(y_1, \ldots, y_n)$ rather than just x_i.
Then the SAT/UNSAT question is similar ($\exists$ values of y_i ...), and the community runs SMT Competitions (`https://smt-comp.github.io/2020/`), but a separate track for each theory, as the problems will be different.
The SMTLIB format [BFT17] provides a standard input format.
UNSAT is in general unsolved (but see [KAED21] for one example).
There is substantial progress in SMT-solving over the years, possibly similar to SAT.

Computer Algebra: where are we

Obviously, Group Theory (etc.) are part of computer algebra: what about the rest?

In general the problems have a bad worst-case complexity, and we want effective solvers for "real" problems, not just "random" ones. The question is "what does this mean?".

- **Format** No common standard. We do have OpenMath [BCC+17], but it's not as widely supported as we would like.
- **Contests** None. Could SIGSAM organise them?
- **Problem Sets** No independent ones. Each author chooses his own.
- **Archive** Not really.

Polynomial g.c.d.

- NP-hard (for sparse polynomials, even univariate) [Pla84].
- Can be challenging for multivariates
- No standard database: trawl previous papers (and often need to ask the authors)
- Verification is a challenge: one can check that the result is *a common divisor*, but verifying *greatest* is still NP-hard [Pla84].

Polynomial Factorisation

- Polynomial-time for dense encodings [LLL82], presumably NP-hard for sparse.
- – No standard database: trawl previous papers (and often need to ask the authors)
- Verification is a challenge: one can check that the result is *a factorisation*, but checking completeness (i.e. that these factors are irreducible) seems to be as hard as the original problem.
- ? With probability 1, a random polynomial is irreducible, so what are the *interesting* problems?

Gröbner Bases

- Doubly exponential (w.r.t. n) worst-case complexity [MR13], even if a prime ideal [Chi09].

\+ There is a collection [BM96]

– Very old (1996) and completely static.

– – Some examples only in PDF.

? No concept of UNSAT, but it's not clear what a certificate might mean.

Real Algebraic Geometry (CAD)

- Doubly exponential (w.r.t. n) worst-case complexity [BD07]

+ There is a collection [Wil14]

– Somewhat old (2014) and completely static.

$\surd$ The DEWCAD project [BDE+21] might update this, but still issues of long-term conservation.

? Format: text, Maple and QEPCAD

? No concept of UNSAT (but see [KAED21]), but it's not clear what a certificate might mean.

Integration

- Complexity is essentially unknown (but certainly involves g.c.d., factorisation etc.)
- A new question here is the "niceness" of the output.
- "Paper" mathematics produced large databases, e.g. [GR07].
  - PDF, and the devil to scan.
- Current best database is described in [JR10].
- Algorithm-based software (e.g. [Dav81]) has an internal proof of UNSAT, but I know of no software that can exhibit it.

Conclusions

1. The field of computer algebra really ought to invest in the sort of contests that have stimulated the SAT and SMT worlds.
2. This requires much larger databases of "relevant" problems than we currently have, and they need to be properly curated.
   + Technology, e.g. wikis, or GitHub, has greatly advanced since [BM96].
3. This would allow much better benchmarking technology [BDG17].

Bibliography I

D. Achlioptas and Y. Peres.
The threshold for random k-SAT is $2^k \log 2 - O(k)$.
J. Amer. Math. Soc., 17:947–973, 2004.

D. Achlioptas and Y. Peres.
Random k-SAT: Two Moments Suffice to Cross a Sharp Threshold.
SIAM J. Comput., 36:740–762, 2006.

T. Balyo and L. Chrpa.
Using algorithm configuration tools to generate hard SAT benchmarks.
In *Eleventh Annual Symposium on Combinatorial Search*, 2018.

Bibliography II

URL: https://algo2.iti.kit.edu/balyo/papers/socs18.pdf.

S. Buswell, O. Caprotti, D.P. Carlisle, M.C. Dewar, M. Gaëtano, M. Kohlhase, J.H. Davenport, and P.D.F. Ion.
The OpenMath Standard 2.0 Revision 1.
http://www.openmath.org, 2017.

C.W. Brown and J.H. Davenport.
The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition.
In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 54–60, 2007.

Bibliography III

R.J. Bradford, J.H. Davenport, M. England, A. Sadeghimanesh, and A. Uncu.
The DEWCAD Project: Pushing Back the Doubly Exponential Wall of Cylindrical Algebraic Decomposition.
https://arxiv.org/abs/2106.08740, 2021.

M.N. Brain, J.H. Davenport, and A. Griggio.
Benchmarking Solvers, SAT-style.
SC^2 2017 Satisfiability Checking and Symbolic Computation CEUR Workshop, 1974(RP3):1–15, 2017.
URL: http://ceur-ws.org/Vol-1974/RP3.pdf.

Bibliography IV

C. Barrett, P. Fontaine, and C. Tinelli.
The SMT-LIB Standard: Version 2.6.
http://smtlib.cs.uiowa.edu/papers/smt-lib-reference-v2.6-r2017-07-18.pdf, 2017.

R.E. Bixby.
Computational Progress in Linear and Mixed Integer Programming.
Presentation at ICIAM 2015, 2015.
URL: http://staff.bath.ac.uk/masjhd/Others/Bixby2015a.pdf.

G. Butler and J. McKay.
The transitive groups of degree up to 11.
Comm. Algebra, 11:863–911, 1983.

Bibliography V

D. Bini and B. Mourrain.
Polynomial test suite.
http://www-sop.inria.fr/saga/POL/, 1996.

G. Butler.
The transitive groups of degree fourteen and fifteen.
J. Symbolic Comp., 16:413–422, 1993.

J.J. Cannon and D.F. Holt.
The transitive permutation groups of degree 32.
Experiment. Math., 17:307–314, 2008.

A.L. Chistov.
Double-exponential lower bound for the degree of any system of generators of a polynomial prime ideal.
St. Petersburg Math. J., 20:983–1001, 2009.

Bibliography VI

S.A. Cook.
On the minimum computation time of functions.
PhD thesis, Department of Mathematics Harvard University, 1966.

J.H. Davenport.
On the Integration of Algebraic Functions, volume 102 of *Springer Lecture Notes in Computer Science*.
Springer Berlin–Heidelberg–New York (Russian ed. MIR Moscow 1985), 1981.

M. Daquino, S. Peroni, D. Shotton, G. Colavizza, B. Ghavimi, A. Lauscher, P. Mayr, M. Romanello, and P. Zumstein.
The OpenCitations Data Model.
https://arxiv.org/abs/2005.11981, 2020.

Bibliography VII

I.S. Gradshteyn and I.M. Ryzhik.
Table of Integrals, Series and Products 7th edition (ed. A. Jeffrey and D. Zwillinger).
Academic Press, 2007.

I.P. Gent and T. Walsh.
The SAT phase transition.
ECAI, 94:105–109, 1994.

A. Hulpke.
Konstruktion transitiver Permutationsgruppen.
PhD thesis, RWTH Aachen, 1996.

A. Hulpke.
Constructing transitive permutation groups.
J. Symbolic Comput., 39:1–30, 2005.

Bibliography VIII

D.J. Jeffrey and A.D. Rich.
Reducing Expression Size Using Rule-Based Integration.
In S. Autexier *et al.*, editor, *Proceedings CICM 2010*, pages 234–246, 2010.

G. Kremer, E. Ábrahám, M. England, and J.H. Davenport.
Cylindrical Algebraic Coverings for Satisfiability Modulo Theories Solving.
In preparation, 2021.

W. Küchlin and C. Sinz.
Proving Consistency Assertions for Automotive Product Data Management.
J. Automated Reasoning, 24:145–163, 2000.

Bibliography IX

J. Kratz and C. Strasser.
Data publication consensus and controversies (version 3).
F1000Research Article 94, 3, 2014.

A.K. Lenstra, H.W. Lenstra Jun., and L. Lovász.
Factoring Polynomials with Rational Coefficients.
Math. Ann., 261:515–534, 1982.

H. Mooney and M.P. Newton.
The Anatomy of a Data Citation: Discovery, Reuse, and Credit.
Journal of Librarianship and Scholarly Communication Article p.eP1035, 1, 2012.

Bibliography X

E.W. Mayr and S. Ritscher.
Dimension-dependent bounds for Gröbner bases of polynomial ideals.
J. Symbolic Comp., 49:78–94, 2013.

D.A. Plaisted.
New NP-Hard and NP-Complete Polynomial and Integer Divisibility Problems.
Theor. Comp. Sci., 31:125–138, 1984.

G.F. Royle.
The Transitive Groups of Degree Twelve.
J. Symbolic Comp., 4:255–268, 1987.

Bibliography XI

N.J.A. Sloane.
The Online Encyclopedia of Integer Sequences.
Notices A.M.S., 50:912–915, 2003.

I. Spence.
Weakening Cardinality Constraints Creates Harder Satisfiability Benchmarks.
J. Exp. Algorithmics Article 1.4, 20, 2015.

The GAP Group.
GAP — Groups, Algorithms, and Programming, Version 4.11.1.
`https://www.gap-system.org`, 2021.

Bibliography XII

S. van de Sandt, L.H. Nielsen, A. Ioannidis, A. Muench, E. Henneken, A. Accomazzi, C. Bigarella, J.B.G. Lopez, and S. Dallmeier-Tiessen.
Practice meets Principle: Tracking Software and Data Citations to Zenodo DOIs.
https://arxiv.org/abs/1911.00295, 2019.

D.J. Wilson.
Advances in Cylindrical Algebraic Decomposition.
PhD thesis, University of Bath, 2014.