Articles and Scheduling for Student Seminar in Combinatorics: Mathematical Software

Komei Fukuda
Department of Mathematics and
Institute of Theoretical Computer Science, ETH Zurich, Switzerland
http://www.inf.ethz.ch/personal/fukudak/

October 17, 2014

1 Seminar Schedule

The schedule is meant to be tentative and may be modified depending on the progress of our seminar. Students may use blackboards, overhead projector, or beamer for presentation. The lecture room is HG E 33.3 (Tuesday 10-12).

Date	Theme	Presenter(s)
September 16	overview, initial planning	Komei Fukuda
September 23	fixing teams and planning	Komei Fukuda
September 30	preparation period	no seminar
October 7	preparation period	no seminar
October 14	GNU GMP	Diyora Salimova
October 21	Convex Hull (cddlib, lrslib, ppl)	Niklas Pfister
October 28	Triangulations (TOPCOM)	Clemens Pohle
November 4	Platform for Polytopes (Polymake)	Sonja Meier, Heidi Pang
November 11	Sage Introduction	C. Schneebeli, M. Göggel, A. Lobbe
November 18	Sage Implementation	C. Schneebeli, M. Göggel, A. Lobbe
November 25	Integer Programming/Lattice Points	Christoph Glanzer
December 2	Group Theory and Symmetries	Loreno Heer
December 9	Theorem Prover	Patrick Schnider
December 16	Polynomials and Gröbner Bases	Alice Feldmann

2 Presentations and Teams

In this seminar, we plan to study some of the most exciting developments of mathematical software, such as those presented ?t International Congress of Mathematical Software (ICMS) [**24, 20, 25, 19**]. We mainly focus on free open-source softwares.

We will have up to 10 teams, each of which consists of one or two students. A team of two can be made with mutual consensus, but we will have alternative ways to make a team by random selection. Each presenter gives a talk on the assigned software system or library for 45 to 90 minutes. The presenter can use any publication on the software, typically the user/developer manual, tutorials or research papers. Our eventual goal is to understand how one can write useful and reliable mathematical software systems or libraries.

3 Themes

The themes we are going to study are flexible in principle. Yet, to make a reasonable plan, I will list recommended themes for the seminar.

GNU GMP To write any (rigorous) mathematical software, it is essential that integer arithmetic is done correctly no matter how large numbers are. Also, for any rational numbers, we should be able to perform basic operations correctly and efficiently. GNU GMP [8] is one of the most reliable and efficient arithmetic libraries available.

Convex Hull One of the most fundamental problem in computational geometry is to compute the convex hull of a finite point set in $\mathbb{Q}^d$. It is also known as the representation conversion between V-representation and H-representation, where V means vertices (or points) and H means halfspaces (or inequalities). Two software libraries cddlib [2] and lrslib [11] are written to perform this conversion with the same input format. Both rely on GMP's exact integer and rational arithmetic, while cddlib can run on floating point arithmetic for faster but non-exact computation. The Parma Polyhedra Library (PPL) [14] is a more recent implementation of the incremental algorithm that cddlib employs and is known to be very efficient.

Triangulations A triangulation of a point set S in $\mathbb{R}^d$ is a special decomposition of the convex hull of S into simplices using the points in S as vertices. Clearly there are only a finite number of triangulations. The basic question is to count the number of triangulations of a point set, for example, the set of 0/1 points in $\mathbb{R}^d$ (the vertices of a d-cube). The software TOPCOM [17, 26] ("a package for computing Triangulations Of Point Configurations and Oriented Matroids") enumerates all triangulations of a point set by using only the combinatorial information (i.e. its oriented matroid) of the point set.

Polymake Polymake [13, 21, 22] provides an integrated environment to do many fundamental computations with convex polytopes, by integrating libraries and codes like cddlib, lrslib and TOPCOM. One can also visualize polytopes and graphs with external programs like javaview.

Sage Sage [15] [20, page 12-27] is a free and open-source software system that has similar and sometimes superior functionalities to commercial Mathematica, Maple and Matlab. For example, it integrates the software package called Gfan [7] that enumerates all reduced Gröbner bases of a polynomial ideal. Commercial programs can compute one reduced basis given a fixed monomial (term) order but not all.

Integer Programming/Lattice Points Counting and Enumerating all lattice points in a convex polytope is a fundamental problem. Finding a Hilbert basis of a rational cone is another hard problem which has applications in integer programming. There are a few strong efforts in implementing algorithms for these problems recently, see [10, 1, 12]

Group Theory and Symmetries Symmetries are a very common thing in mathematics. By exploiting symmetries of the underlining mathematical structure one can make a computer program exponentially faster. GAP [6] is "a system for computational discrete algebra, with particular emphasis on Computational Group Theory." It can be used as a preprocessing of input so that the main computation avoids generating redundant duplicates with respect to the underlying symmetries.

Polynomials and Gröbner Bases Creating efficient implementations of Buchberger's algorithm for Gröbner bases was one strong driving force of the recent advancement of computational algebra. Singular [16] and CoCoA [3] are two free open source software systems in this direction. Gfan [7] is a unique program that can enumerate all reduced Gröbner bases.

Theorem Prover Generating a rigorous (formal) proof for a mathematical theorem is a frequently discussed area of computational mathematics. One project which attracted an enormous attention is to formalize Tom Hales' proof of the Kepler conjecture, see [20, page 1–3], [24, page 16–20], using the HOL Light theorem prover [9]. Another theorem prover which was used to formalize [23] a proof of the 4 colour theorem is Coq [4].

4 Computational Environment

Everyone taking this course must have an access to a computational environment which allows one to compile and execute publicly available programs written in commonly used computer languages. On **linux** systems, this is not a problem as long as basic developer tools (e.g. C/C++ compilers) are installed properly. On Apple OSX operating systems, one must install the (free) developer tools known as **Xcode** developer tools [18]. On both linux and macos systems, installation can be done only by superuser (root) or any user with administrator status. On Windows, one must install the (free) **Cygwin** unix-like environment [5] and developer tools and libraries.

5 Final Report

Each student (not a team) must submit a final report in pdf of 5 to 10 pages written in latex covering the presented material, detailed descriptions and possibly your comments/ideas, within four weeks after the presentation.

6 Articles Online

In addition to making use of web search engines, each student is expected to learn to use the AMS (American Mathematical Society) **MathSciNet** database to search for articles you wish to read: http://www.ams.org/mathscinet/index.html . Even if the database item of the article has no link to the paper in pdf, please do not give up. Go to the journal site and search there for the pdf. By now, most of the important articles are available online. Please make sure that your network connection is established within the ETH domain (by possibly using VPN from home).

References

[1] **4ti2** —a software package for algebraic, geometric and combinatorial problems on linear spaces. `www.4ti2.de`.

[2] **cdd**, **cddplus** and **cddlib** homepage. `http://www.inf.ethz.ch/personal/fukudak/cdd_home/`.

[3] **CoCoA** — Computations in commutative algebra. `http://cocoa.dima.unige.it`.

[4] **Coq** — a formal proof management system. `http://coq.inria.fr`.

[5] **Cygwin** for windows. `https://www.cygwin.com`.

[6] **GAP** — groups, algorithms, programming. `http://www.gap-system.org`.

[7] **Gfan**. `http://home.imf.au.dk/jensen/software/gfan/gfan.html`.

[8] **GMP**, GNU's library for arbitrary precision arithmetic. `http://gmplib.org/`.

[9] **HOL Light**. `http://www.cl.cam.ac.uk/~jrh13/hol-light/`.

[10] **lattE**. `https://www.math.ucdavis.edu/~latte/`.

[11] **lrs** homepage. `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`.

[12] **normaliz**. `http://www.home.uni-osnabrueck.de/wbruns/normaliz/`.

[13] **polymake**. `http://www.polymake.org/doku.php`.

[14] **PPL**, the parma polyhedra library. `http://bugseng.com/products/ppl/`.

[15] **Sage**. http://www.sagemath.org.

[16] **Singular**. www.singular.uni-kl.de/.

[17] **TOPCOM**. http://www.rambau.wm.uni-bayreuth.de/TOPCOM/.

[18] **Xcode** developer tools. https://developer.apple.com/xcode/downloads/.

[19] Arjeh M. Cohen, Xiao-Shan Gao, and Nobuki Takayama, editors. *Mathematical software*. World Scientific Publishing Co., Inc., River Edge, NJ, 2002.

[20] Komei Fukuda, Joris van der Hoeven, Michael Joswig, and Nobuki Takayama, editors. *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*. Springer-Verlag, 2010. http://link.springer.com/book/10.1007/978-3-642-15582-6.

[21] E. Gawrilow and M. Joswig. polymake: a framework for analyzing convex polytopes. In *Polytopes—combinatorics and computation (Oberwolfach, 1997)*, volume 29 of *DMV Sem.*, pages 43–73. Birkhäuser, Basel, 2000. http://link.springer.com/book/10.1007/978-3-0348-8438-9.

[22] E. Gawrilow and M. Joswig. Flexible object hierarchies in polymake (extended abstract). In *Mathematical software—ICMS 2006*, volume 4151 of *Lecture Notes in Comput. Sci.*, pages 219–221. Springer, Berlin, 2006. http://dx.doi.org/10.1007/11832225_20.

[23] G. Gonthier. A computer-checked proof of the four colour theorem. http://research.microsoft.com/en-us/um/people/gonthier/4colproof.pdf.

[24] Hoon Hong and Chee Yap, editors. *Mathematical Software - ICMS 2014*, volume 8592 of *Lecture Notes in Computer Science*. Springer-Verlag, 2014. http://link.springer.com/book/10.1007/978-3-662-44199-2.

[25] Andrés Iglesias and Nobuki Takayama, editors. *Mathematical software—ICMS 2006*, volume 4151 of *Lecture Notes in Computer Science*. Springer, Berlin, 2006. http://link.springer.com/book/10.1007/11832225.

[26] J. Rambau. TOPCOM: triangulations of point configurations and oriented matroids. In *Mathematical software (Beijing, 2002)*, pages 330–340. World Sci. Publ., River Edge, NJ, 2002. http://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/684.