

Computer Algebra – use a computer to solve difficult maths problems!

Lars Åström¹,
Supervisor: Victor Ufnarovski¹

¹Department of Mathematics, Lund University

In mathematics it is not only important to solve a problem, but in fact the proof of correctness is usually just as important. Proving equalities turns out to be quite hard sometimes, which is why a automatic prover is highly useful!

Computer algebra is a field in mathematics where a computer is used to manipulate mathematical expressions, for instance add, multiply and divide polynomials. There are many existing softwares that do this, such as Maple and Mathematica, but even the simplest operations turn out to be quite tricky to implement.

The thesis produced a software library which can prove a specific type of equations. The software works by getting an equation as input and then produces a formal proof for that the equation actually is true. In the proof process an method called Wilf-Zeilberger's method (Wilf, 1994) is used to produce the intermediate steps.

Combinatorics is a part of mathematics concerned with counting, for instance "In how many ways can one pick a three cookies out of five different types of cookies?". In combinatorics one often encounters the problem of summation, and proving that two expressions are equal. This can be solving the question "In how many ways can one pick cookies (any number) out of five different types of cookies?". This can be done by adding the number of solutions if we pick exactly 0, 1, 2, 3, 4 and 5, respectively. We can also solve the problem by noting that we have two choices for each cookie, to take it or not. Since we have five cookies the total number of choices becomes $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^5$. Therefore we should have that

$$\sum_{i=0}^5 \text{Ways to pick } i \text{ out of 5 cookies} = 2^5.$$

By computing the left and right side of the equation one can verify that equality occurs, but can this result be generalized?

The program produced in this thesis addresses this problem and can in an automatic way produce a proof of an equality. The program outputs a proof, where the user only need to verify a few statements – which usually is an easy task. This provides a great help for a user in the process of proving an equality.

Producing an automatic equation prover required a few steps – such as reading the equation that should be proved, running the method and printing the proof in the right format. The most important point I learned was the importance of continuous testing. Every single part of the program was tested both individually and as a part of the bigger picture in the program. This resulted in that as much as 30% of the almost 2300 lines of code that were written were purely for testing.

Producing mathematical proofs automatically is quite cool, and indeed the main contribution of the thesis. Still one major thing that came out of the thesis work was the understanding of Computer Algebra, especially how something can seem easy but is terribly hard to implement. The number of times I underestimated the time to implement something was uncountable.

References

1. Veltman, M.J.G. and Williams, D.N. (1991). 'Schoonschip '91.'
2. Lopez, R. *Computer Algebra Systems*. Maplesoft, viewed 26 November 2019. <https://www.maplesoft.com/ns/maple/cas/computer-algebra-systems-math-education.aspx>.
3. Güyer, T. (2008). 'Computer Algebra Systems as the Mathematics Teaching Tool.' *World Applied Sciences Journal* 3 (1): 132-139, 2008.
4. Wilf, H.S., and Zeilberger, D. (1990). 'Rational functions certify combinatorial identities.' *J. Amer. Math. Soc.* 3 (1990), pp. 147–158.
5. Paule, P. and Schorn, M. (1994). 'A Mathematica Version of Zeilberger's Algorithm for Proving Binomial Coefficient Identities.' *J. Symbolic Computation* (1995) 20, pp. 673–698.
6. Gosper, R.W. Jr. (1978). *Decision procedure for indefinite hypergeometric summation*. Proc. Natl. Acad. Sci. USA. Vol. 75, No. 1, pp. 40–42, January 1978. Palo Alto, California.
7. Wilf, H.S. (1994). *Generating functionology*. Academic Press, Inc. Philadelphia, Pennsylvania.
8. Johnson, M. (2007). *Handout 08 Bottom-up Parsing*. Lecture notes Stanford University CS143. Delivered 2 July 2007.
9. Gould, H.W. (1972). *Combinatorial Identities*. Morgantown Printing and Binding Co. West Virginia.
10. Tesler (2017). *Chapter 3.3, 4.1, 4.3. Binomial Coefficient Identities*. Lecture notes University of California San Diego Math 184A. Delivered winter 2019.