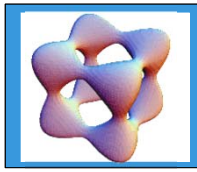
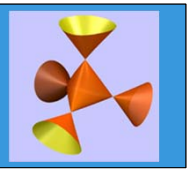


Measuring Improvements to Gröbner Basis Computation with S-polynomials

John Wallace
Advisor: Professor Takunari Miyazaki



Introduction

Gröbner bases are finite sets of polynomials satisfying a set of properties that make them useful in solving a variety of mathematical problems. In Computer Science, Gröbner bases are of great interest due to the intrinsic difficulty of converting some finite set of polynomials into a Gröbner basis: a problem in the double exponential complexity class. However, unlike problems of comparable difficulty, there is a wide spectrum of algorithms that have been designed to solve the problem.

In practice, the efficiency of these algorithms is measured by timing them. However, time captures every operation in an algorithm, including pre-processing and operations whose cost is unlikely to scale significantly with input size. In this project we investigated counting the number of times the most costly step of an algorithm was performed as a way of measuring the algorithm's efficiency. Computing what is known as an S-polynomial has widely been identified (2. and 3.) as the most costly step in most Gröbner basis algorithms, and is what was used in this project as the metric against which algorithms were compared. In total, six algorithms were implemented in the computer algebra system SINGULAR and were compared using this method.

Leading Terms

Def. The **leading term** of a multivariable polynomial, p, denoted LT(p) is the term of p with highest degree, where ties are broken by the rule $x > y > z$.

Ex. $p(x,y,z) = x^2y^2z+x^2yz^2+z$.

The polynomial p has two terms of degree five: x^2y^2z and x^2yz^2. To decide which term is leading, we look at the powers of individual variables using the rule $x > y > z$. Both x^2y^2z and x^2yz^2 have the same power of x, (i.e. 2), but x^2y^2z has a higher power of y than x^2yz^2 so x^2y^2z is the leading term of p.

Gröbner Bases

Def. A Gröbner basis is a finite set of polynomials, G, such that the leading term of any sum of multiples of polynomials in G is divisible by the leading term of some polynomial in G.

Ex. Consider the set of polynomials $F=\{x+y, 2x-y\}$. A particular combination of these polynomials is the polynomial

$$p(x,y)=(2)(x+y)-(2x-y)=3y.$$

Clearly, the leading term of p is $3y$, but this leading term is not divisible by either of the leading terms of polynomials in F, i.e. x and $2x$. Therefore F is not a Gröbner basis. It can be checked however that $F'=\{x+y, 2x-y, 3y\}$ is a Gröbner basis.

Note that the counter example used to show that F was not a Gröbner basis, i.e. $3y$, was used to turn F into a Gröbner basis by adding it to F. This strategy for turning sets into Gröbner bases is captured by S-polynomials.

S-polynomials

Def. An **S-polynomial** of two polynomials f and g is

$$S(f,g) = \frac{\mathrm{LCM}(\mathrm{LT}(f),\mathrm{LT}(g))}{\mathrm{LT}(f)} \cdot f - \frac{\mathrm{LCM}(\mathrm{LT}(f),\mathrm{LT}(g))}{\mathrm{LT}(g)} \cdot g$$

S-polynomials are defined specifically to cancel the leading terms of f and g yielding a polynomial with a potentially new leading term. A set is a Gröbner basis if and only if $S(f,g)=0$ for all f and g.

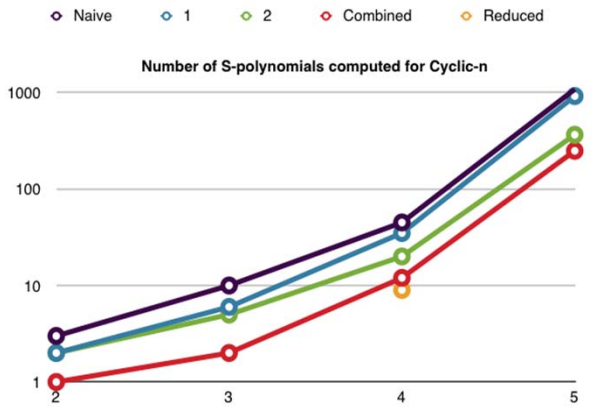
S-polynomial Data and Analysis

Implementation: Our algorithms were written in the computer algebra system SINGULAR. Of the five algorithms first implemented, one is 'naive' (computes every S-polynomial) and acted as a kind of control.

S-polynomial Step: Each algorithm recorded the number of times it computed an S-polynomial as well as the number of times an S-polynomial calculation was avoided.

Data: Over 50 problem instances were used, a large portion of which were gathered from our sources (1-4). In most cases the improved algorithms succeeded in computing fewer S-polynomials. The best results were found when three of the algorithms were integrated into a single *combined* algorithm.

Analysis: The success of the combined algorithm is consistent with the analyses given in 2. and 3. What was not present in their analyses, and what S-polynomial analysis revealed was that one of the improved algorithms did the lion's share of the work within the combined algorithm. This phenomenon, as well as the overlap between improvements within the combined algorithm isn't at all captured by time and presents an advantage of this approach.



Reduced Gröbner Bases

S-polynomials Revisited: Instead of analyzing the total number of S-polynomials computed, we can also analyze the ratio of useful S-polynomials to the total number computed. This ratio depends on the total number of S-polynomials computed, as well as the number of elements in the resulting basis.

Algorithm: A *reduced Gröbner basis* algorithm outputs a smallest possible Gröbner basis. Such an algorithm stands to do well in terms of the S-polynomial ratio.

Results: The reduced algorithm we implemented showed improvements over the naive algorithm in both senses. However when compared to the combined algorithm there were instances in which it was more efficient but did more work, leading to a discrepancy between the two concepts (and an interesting question of when to use which). Moreover this algorithm was very slow, which is a limitation of this approach.

Future Research

- Use S-polynomial computations (or the analogous operation) for measuring the performance of the many other Gröbner basis algorithms.
- Including other operations (such as polynomial reductions), in addition to S-polynomial computations, to measure algorithmic performance.
- Implementation specific improvements to the combined algorithm and the reduced Gröbner basis algorithm so they can be compared on more difficult problem instances.

Acknowledgements

I would like to thank Professors Takunari Miyazaki and Madalene Spezialetti of the Trinity College Computer Science Department, and Professor Kirsti Wash of the Trinity College Mathematics Department for their advising and guidance this project. I'd also like to recommend SINGULAR and its libraries.

Works Cited

1. Grassman et. al. On an Implementation of Standard Bases and Syzygies in SINGULAR, AAECC (1996) 7: 235.
2. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, Multidimensional Systems Theory (1985).
3. Cox, Little, O'Shea. Ideals Varieties, and Algorithms (Fourth Edition). Springer Undergraduate Texts in Mathematics (2015).
4. Giovani et. al. "One Sugar Cube, Please" or Selection Strategies in the Buchberger Algorithm. Proceedings of the 1991 ISSAC, 55-63.