MEASURING IMPROVEMENTS TO GRÖBNER BASIS COMPUTATION WITH S-POLYNOMIALS

John Wallace ‘17
Faculty Sponsor: Takunari Miyazaki

Gröbner bases are finite sets of polynomials satisfying a set of properties that make them widely applicable in a variety of fields including computational mathematics, computer algebra, and mathematical modeling. The problem of computing Gröbner bases is, in general, a very difficult problem and one of the most frequently cited examples of a problem lying in the double-exponential complexity class. Virtually all of the algorithms designed for improving the efficiency of Gröbner basis computation have sought to do so by reducing the number of S-polynomials computed throughout computation. An S-polynomial is a combination of two polynomials critical to computing a Gröbner basis; the computation of which has been widely identified as the “computational bottleneck” of the problem. Despite this, algorithms are mostly compared by timing their performance on a number of problem instances.

In this project, the number of S-polynomials computed was used to compare the efficiency of Gröbner basis algorithms, as opposed to timing the algorithms on like inputs. This approach was taken in order to isolate the most expensive aspect of Gröbner basis computation from other algorithmic operations that are less fundamental to the problem (but still captured when measuring performance with time). To accomplish this comparison, we implemented five Gröbner basis algorithms in the computer algebra system SINGULAR and recorded the number of S-polynomials computed as well as the number of S-polynomials avoided. The results show, quite decidedly, that two of these algorithms are superior to the others. Interestingly, S-polynomial comparisons fail to distinguish which of these algorithms is more efficient, due in part to one algorithm being very inefficient with respect to time. Moreover, comparing these two algorithms gave rise to two distinct ways of using S-polynomials to measure algorithmic efficiency.