The Skeletons you find when you Order your Ideal's Closet

John Perry
University of Southern Mississippi (United States)

john.perry@usm.edu

Abstract

The best-known algorithms to compute a Gröbner basis of an ideal are *static*, in that they require a term ordering as input, and return a Gröbner basis with respect to that same ordering. The choice of ordering can have a critical effect on the time and memory consumed, and while some orderings have a reputation for efficiency, they are not always a good choice, let alone the best choice.

A *dynamic* algorithm to compute a Gröbner basis does not require a term ordering as input, but tries to compute an efficient term ordering *as the computation proceeds*, returning both a final ordering, and a Gröbner basis with respect to that ordering. Past implementations have relied on linear programming to do this, but the associated linear programs grow very large, very quickly, putting many ideals of interest out of this technique's reach.

The solution set to these linear programs takes the form of a polyhedral cone, and a dynamic algorithm can use the cone's skeleton to reduce drastically both the size and number of linear programs, making it possible to compute ideals that were previously beyond the dynamic algorithm's practical reach. This talk, based on joint work with Massimo Caboara of the University of Pisa, reviews the idea of a dynamic algorithm, describes the new technique, and demonstrates its benefits.

Keywords

Gröbner bases, dynamic Buchberger algorithm, polyhedral cones, linear programming, term ordering