PARALLELIZATION OF THE BUCHBERGER ALGORITHM

A. A. Letichevskii

INTRODUCTION

Buchberger's algorithm for the construction of Grobner bases of polynomial ideals [1] is one of the most popular algorithms of modern computer algebra. This is an interesting algorithm in many respects. Its practical value is determined by its applicability to solution of systems of nonlinear algebraic equations and also to many problems of algebraic and computational geometry. It is an effective example of a CPC (critical-pair completion) algorithm [2]. Its computational complexity (the known bounds are superexponential) forces us to look for parallel implementations.

The parallelization of the Buchberger algorithm is a fairly difficult problem, as the algorithm is highly irregular. This problem has been tackled by a number of authors [3-7], who mainly provide a descriptive presentation of various implementations of the algorithm in systems with shared and distributed memory. We start our article with a review of the declarative representation of the Buchberger algorithm as a system of rewrite rules. This representation may serve as a source for various sequential and parallel implementations. Some of these implementations are discussed. Results of model experiments are presented for a program developed in the APS system [8, 9], which uses the rewrite technique as the main computational mechanism [10].

GROBNER BASES

Consider the ring \( P = \mathbb{K}[x_1, ..., x_n] \) of polynomials in the variables \( x_1, ..., x_n \) over the ring \( \mathbb{K} \). The set \( F = \{ f_1, ..., f_m \} \) of polynomials from this ring generates the ideal \( I(f_1, ..., f_m) \) that consists of all possible polynomials of the form \( \sum_{i=1}^{m} A_i f_i \), where \( A_i \) are arbitrary polynomials from the ring \( P \). Grobner bases are mainly constructed to decide if an arbitrary polynomial \( u \) is contained in the ideal \( I = I(f_1, ..., f_m) \). This is equivalent to the problem of deciding that the relationship \( u = 0 \) is a consequence of the relationships \( f_1 = 0, ..., f_m = 0 \), and also solves the identity problem in the factor-ring of the ring \( P \) by the ideal \( I(u = v \mod I), u - v \in I \).

The decision problem associated with the inclusion \( u \in I \) can be solved by simplifying the polynomial \( u \) using the relationships \( f_1 = 0, ..., f_m = 0 \). To this end, we linearly order the products of powers of the variables \( x_1, ..., x_n \) by the relationship \( \leq \) (e.g., lexicographically). Any ordering can be used that satisfies the conditions \( 1 \leq u \) and \( u \leq v = u g \leq v g \) for an arbitrary product \( g \) of powers of the variables. Each polynomial is a sum of monomials (products of powers of variables with nonzero coefficients), and ordering the summands we obtain a canonical representation of polynomials corresponding to the given ordering of products of powers of the variables. In particular, each polynomial is uniquely representable in the form \( u = v + w \), where \( v \) is the highest order monomial (i.e., the monomial with the greatest product of powers of the variables). Since multiplication of a polynomial by an element of the field \( \mathbb{K} \) does not affect its membership in the ideal, each polynomial can be normalized by dividing all its coefficients by the coefficient of the highest order monomial. We accordingly assume that this coefficient is always 1.

Represent each of the polynomials \( f_1, ..., f_m \) as the sum \( f_i = g_i + h_i, i = 1, ..., n \), where \( g_i \) is the highest order monomial of the polynomial \( f_i \). Consider the system of relationships (reductions) \( v g_i + x = x - v h_i \), where \( v \) is an arbitrary...
monomial and $x$ an arbitrary polynomial, $i = 1, \ldots, n$. It is easy to see that $v g_i + x \in I \leftrightarrow x - v h_i \in I$. Therefore, if the polynomial $u$ is representable in the form $u = v g_i + x$, it can be transformed by the reduction relationship to the form $x - v h_i$. If $v g_i$ is the highest order monomial of the polynomial $u$, then its highest order monomial is decreased by reduction. Therefore, the process of reduction, i.e., the application of reduction relationships to the polynomial $u$, will stop after finitely many steps, when it produces a polynomial irreducible by the set $F$. If the polynomial $u$ is reducible to zero, then $u \in I$. The converse, in general, is not true.

The Grobner basis $G$ is defined as the set of polynomials for which the end result of reduction of any polynomial is defined uniquely apart from a scalar multiplier, i.e., is independent of the order of the reduction. If $G$ is a Grobner basis and $v$ the result obtained by reduction of the polynomial $u$, then $u \in I(G) \leftrightarrow v \in I(G)$. The problem thus reduces to constructing for an arbitrary set of polynomials $F$ the Grobner basis $G$ that generates the same ideal as $F$ itself.

Buchberger's method for the construction of the Grobner basis for the set of polynomials $F$ completes the given set with missing polynomials. These polynomials are sought among polynomials of the form $Af + A'f'$, where $f, f' \in F$, and the coefficients are selected so that the highest order monomials of the addends are reduced. Let $f = g + h, f' = g' + h'$, where $g$ and $g'$ are the highest order monomials. Find the least common multiple of $g$ and $g'$, i.e., the least normalized monomial (product of powers of the variables) lcm$(g, g')$ such that lcm$(g, g') = ug = u'g'$, and consider the polynomial $Spl(f, f') = uf - u'f' = uh - u'h'$ (the $S$-polynomial). If this polynomial is not reducible to zero, then it may be added to the set $F$ (the current basis). It is easy to show that the completion procedure always ends. Indeed, after reduction the highest order monomial of the nonzero $S$-polynomial is always relatively prime with all the elements of the current basis, and by Dixon's well-known lemma there are no infinite sequences of relatively prime normalized monomials of a finite number of variables. On the other hand, reducibility to zero of each $S$-polynomial constructed using the given set of polynomials is a characteristic property of Grobner bases (Buchberger's theorem), and the completion procedure produces one such basis.

ALGEBRAIC REPRESENTATION OF THE ALGORITHM

The Buchberger algorithm is usually represented as a procedure that reduces polynomials by the current basis and constructs $S$-polynomials if it can be established in advance that they are reducible to zero. The procedural representation is not easily amenable to direct parallelization. We therefore use a more declarative representation of the algorithm as a system of rewrite rules. This is the representation used in the APLAN implementation of the Buchberger algorithm in the APS system.

As the basic operations we use the following predicates and functions:

- the predicate $Div(x, y)$ determines the divisibility of the highest order monomial of the polynomial $x$ by the highest order monomial of the polynomial $y$;
- the predicate $x < y$ equals 1 if the highest order monomial of the polynomial $x$ is less than the highest order monomial of the polynomial $y$;
- the function $reduce(x, y)$ reduces the polynomial $x$ by the polynomial $y$, i.e., applies to the polynomial $x$ the reduction relationship corresponding to the polynomial $y$ as long as this is possible; the reduction result is normalized;
- the function $Spl(x, y)$ computes the $S$-polynomial for the polynomials $x$ and $y$.

The main data structure for the construction of the Grobner basis is the list $G$ of all polynomials participating in the process of construction. These polynomials are subdivided into several types:

- initial pretenders, as yet unreduced polynomials, which include the original elements of the set $F$ and the $S$-polynomials that may be chosen to complete the current basis;
- reduced pretenders, i.e., polynomials being processed in the current reduction loop by the current basis;
- old pretenders, i.e., polynomials that have completed the current reduction loop and must be submitted to another loop;
- candidates, i.e., polynomials that have been completely reduced by the current basis;
- current basis elements.

Each polynomial $p$ enters the list $G$ in the form $i:p$, where $i$ is the type index according to the following convention: 1 — initial pretenders; 2 — reduced pretenders; 3 — old pretenders; 4 — candidates.