In *Computer-Aided Verification, CAV '96* (Rajeev Alur and Thomas A. Henzinger, eds.), LNCS vol. 1102, 1991, 26–37, © Springer-Verlag

Symbolic Model Checking Using Algebraic Geometry

George S. Avrunin

Department of Mathematics, University of Massachusetts, Amherst, MA 01003-4515
avrunin@math.umass.edu

Abstract. In this paper, I show that methods from computational algebraic geometry can be used to carry out symbolic model checking using an encoding of Boolean sets as the common zeros of sets of polynomials. This approach could serve as a useful supplement to symbolic model checking methods based on Ordered Binary Decision Diagrams and may provide important theoretical insights by bringing the powerful mathematical machinery of algebraic geometry to bear on the model checking problem.

1 Introduction

Symbolic model checking [8, 13] with Ordered Binary Decision Diagrams (OBDDs), or variants of OBDDs, is a widely used and successful technique for verifying properties of concurrent systems, both hardware and software. But there are many systems for which the OBDDs are too large to make model checking feasible and, aside from a few results like McMillan's theorem on bounded width circuits [13] or Bryant's theorem on integer multiplication [5], there is little theoretical guidance to indicate precisely when the OBDD methods are practical.

It therefore seems worthwhile to investigate alternative "symbolic" representations of Boolean sets that could be used for model checking. Such representations, if they are practical at all, would presumably allow efficient model checking of somewhat different classes of systems than OBDDs, and thus supplement existing symbolic model checking methods. Furthermore, an alternative representation might lead to new theoretical insights into the practicality of symbolic model checking, thereby providing guidance to system developers choosing methods for verifying properties of their systems. This is especially true if there is already a substantial body of theory concerning the proposed representation.

In this paper, I show how computational algebraic geometry can provide representations of Boolean sets suitable for symbolic model checking. The basic idea is that any Boolean set can be regarded as the common zeros of a finite set of polynomials with coefficients in the field of two elements. Such a set of polynomials then provides a symbolic representation of the Boolean set. For example, the common zeros of the set of polynomials $\{x_1 + x_2 + \cdots + x_n, x_1x_2\}$ are exactly the points $(a_1, \ldots, a_n)$ for which an even number of the a_i are 1, and at least one of x_1 and x_2 is zero (all the arithmetic is done modulo 2). A *Gröbner basis* is a canonical choice of such a set of polynomials, and there exist algorithms for finding the Gröbner basis corresponding to a particular Boolean set and for carrying out, at the level of Gröbner bases, the manipulations of Boolean sets required for model checking. Thus, Gröbner bases can be used for symbolic model checking in essentially the same way that OBDDs are.

Algebraic geometry is the study of the geometric objects arising as the common zeros of collections of polynomials. It is an old and rich area of mathematics, and one in which there has been enormous activity and progress in the last few years. In particular, algebraic geometers have studied questions related to the action of groups of symmetries and to the mappings that correspond to abstraction techniques, and considerable attention has been given to computational issues. An approach to symbolic model checking making use of methods from algebraic geometry therefore seems to have considerable promise, both as a supplement to existing methods and as a way to bring a large body of powerful mathematical machinery to bear on the model checking problem.

In the next two sections, I sketch some of the necessary background in algebraic geometry and Gröbner basis methods. The fourth section briefly illustrates the ideas with a small example, and the last section contains a discussion of some of the directions for further investigation of this approach.

2 Some Algebraic Geometry

This section contains an extremely brief presentation of the algebraic geometry needed in the sequel. Any standard text will provide the details and proofs omitted here; the interested reader might consult, for example, the books by Cox, Little, and O'Shea [9] and Hartshorne [11].

We start by setting up some machinery for describing sets of polynomials. Let k be a field (for our applications, k will usually be the field of two elements, the integers modulo 2), and let $k[x_1, \ldots, x_n]$ be the ring of polynomials in the variables $x_1, \ldots, x_n$ with coefficients in k, under the standard addition and multiplication of polynomials. That is, a polynomial is a finite k-linear combination of monomials $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, where the α_i are nonnegative integers, and multiplication of polynomials is defined by setting $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n} \cdot x_1^{\beta_1} x_2^{\beta_2} \ldots x_n^{\beta_n} = x_1^{\alpha_1+\beta_1} x_2^{\alpha_2+\beta_2} \ldots x_n^{\alpha_n+\beta_n}$ and extending linearly to products of arbitrary polynomials. Note that the multiplication is commutative and that the element $1 = x_1^0 x_2^0 \ldots x_n^0$ is an identity element for multiplication.

The basic structure of polynomial rings (or any commutative rings) is given in terms of subsets called ideals. In this setting, ideals are not subrings in general, but they play a role in commutative ring theory analogous to that played by normal subgroups in the theory of groups. An *ideal* is a nonempty subset of $k[x_1, \ldots, x_n]$ that is closed under addition and closed under multiplication by any element of the ring. If $F = \{ f_\alpha \mid \alpha \in \mathscr{A} \}$ is a set of polynomials in $k[x_1, \ldots, x_n]$ indexed by the (not necessarily finite) set $\mathscr{A}$, the *ideal generated by* F is the set of sums of the form $\sum_{a \in \mathscr{A}} h_a f_a$, where the $h_a \in k[x_1, \ldots, x_n]$ and only finitely many of the h_a are nonzero. We will write $\langle F \rangle$ for the ideal generated by F. When $F = \{f_1, \ldots, f_s\}$ is a finite set, we often write $\langle f_1, \ldots, f_s \rangle$ for $\langle F \rangle$, and we say that F is a *basis* for the ideal $\langle f_1, \ldots, f_s \rangle$. The Hilbert Basis Theorem tells us that every ideal in the ring $k[x_1, \ldots, x_n]$ is generated by some finite set of polynomials.

We can think of the polynomials as k-valued functions on the vector space k^n in the usual way: we evaluate $f(x_1, \ldots, x_n)$ at the point $(a_1, \ldots, a_n)$ by substituting a_1 for x_1, a_2 for x_2, and so on. We say that $(a_1, \ldots, a_n)$ is a *zero of* f if $f(a_1, \ldots, a_n) = 0$.

Let F be a (not necessarily finite) subset of $k[x_1, \ldots, x_n]$. The *variety* defined by F, written $\mathbf{V}(F)$, is the set of points in k^n that are zeros of all the polynomials in F. Thus $\mathbf{V}(F) = \{\, (a_1, \ldots, a_n) \in k^n \mid f(a_1, \ldots, a_n) = 0 \text{ for all } f \in F \,\}$.

As usual, if $F = \{f_1, \ldots, f_s\}$ is a finite set, we sometimes write $\mathbf{V}(f_1, \ldots, f_s)$ rather than $\mathbf{V}(F)$. It is not hard to see that $\mathbf{V}(f_1, \ldots, f_m) = \mathbf{V}\left(\langle f_1, \ldots, f_m\rangle\right)$, so we can think of every variety as being the variety defined by some ideal.

If $V_1 = \mathbf{V}(I_1)$ and $V_2 = \mathbf{V}(I_2)$ are the varieties defined by ideals I_1 and I_2, then $V_1 \cap V_2 = \mathbf{V}\left(\langle I_1, I_2\rangle\right)$ and $V_1 \cup V_2 = \mathbf{V}(I_1 \cdot I_2)$, where $I_1 \cdot I_2 = \langle f_1 f_2 \mid f_1 \in I_1, f_2 \in I_2\rangle$. If $I_1 = \langle f_1, \ldots f_r\rangle$ and $I_2 = \langle g_1, \ldots, g_s\rangle$, then $I_1 \cdot I_2 = \langle f_i g_j \mid 1 \leq i \leq r, 1 \leq j \leq s\rangle$.

In general, not every subset of k^n is the variety of some ideal (the varieties are the closed sets of a certain topology on k^n), but each point $(a_1, \ldots, a_n)$ is the variety of the ideal $\langle x_1 - a_1, x_2 - a_2, \ldots, x_n - a_n\rangle$. Since the union of a finite collection of varieties is a variety, any finite set of points is a variety. If k is finite, as will be the case in our application, any subset of k^n is finite, and therefore is a variety.

For the rest of this section, assume that k is the field of two elements.

As just mentioned, we can regard any set of points in k^n as the variety of some ideal. We can then use the ideal, or any basis for the ideal, as a way of encoding the set of points, just as we might use an OBDD. For instance, k^n is the variety of the ideal consisting of the constant polynomial 0, and the empty subset of k^n is the variety of the constant polynomial 1. A somewhat more interesting example is the following.

Choose a positive integer r and let $s = 2^r$. Regard a point $(a_1, \ldots, a_{rs}) \in k^{rs}$ as a list of s numbers between 0 and $s - 1$ by treating each block of r coordinates $a_{ri+1}, a_{ri+2}, \ldots, a_{r(i+1)}$ as the binary representation of a nonnegative integer, and let V be the set of points corresponding to lists in which each number from 0 to $s-1$ occurs exactly once. To construct an ideal I such that $V = \mathbf{V}(I)$, let $f_{i,j}$ be the polynomial $(x_{ri+1} + x_{rj+1} + 1)(x_{ri+2} + x_{rj+2} + 1) \cdots (x_{r(i+1)} + x_{r(j+1)} + 1)$. The polynomial $f_{i,j}$ is zero at a point $(a_1, \ldots, a_{rs})$ if and only if $a_{ri+k} \neq a_{rj+k}$ for some k, so if and only if the ith and jth entries in the list corresponding to $(a_1, \ldots, a_{rs})$ are different integers. Then $V = \mathbf{V}(\, f_{i,j} \mid i < j)$. Other examples are given in Section 4.

Note that there will be more than one ideal I defining a given variety. For instance, the ideals $\{0\}$ and $\langle x_1^2 + x_1, \ldots, x_n^2 + x_n\rangle$ both define the variety k^n (since both 0 and 1 satisfy the equation $x^2 + x = 0$ when we are working modulo 2). In order to do symbolic model checking, we need to be able to determine when two ideals represent the same set of points. We first describe how to do this over a larger field. Let $\bar{k}$ be the algebraic closure of k (this is the smallest extension of k in which every polynomial over k has a root, as every polynomial with coefficients in $\mathbb{R}$ has a root in $\mathbb{C}$). Since $k[x_1, \ldots, x_n] \subseteq \bar{k}[x_1, \ldots, x_n]$, we can regard polynomials in $k[x_1, \ldots, x_n]$ as functions on $\bar{k}^n$, and, for a subset F of $k[x_1, \ldots, x_n]$, we define $\overline{\mathbf{V}}(F)$ to be the points in $\bar{k}^n$ where all the elements of F are zero. For an ideal I, the *radical of* I, denoted by $\sqrt{I}$ is the ideal $\{\, f \in k[x_1, \ldots, x_n] \mid f^s \in I \text{ for some positive integer } s \,\}$. If I_1 and I_2 are ideals of $k[x_1, \ldots, x_n]$, then $\overline{\mathbf{V}}(I_1) = \overline{\mathbf{V}}(I_2)$ if and only if $\sqrt{I_1} = \sqrt{I_2}$. (This is Hilbert's Nullstellensatz.) The Gröbner basis methods described in the next section provide good algorithms for determining when $\sqrt{I_1} = \sqrt{I_2}$, so we can determine when two ideals determine the same variety over the algebraic closure of k.

In general, this does not tell us anything about whether $\mathbf{V}(I_1) = \mathbf{V}(I_2)$, but it does settle the question for a certain class of ideals. Let $Z = \{\, x_i^2 + x_i \mid i = 1, \ldots, n \,\}$. As noted above, every point in k^n is a zero of all the elements of Z, so, for any ideal I, $\mathbf{V}(I) = \mathbf{V}(I) \cap \mathbf{V}(Z) = \mathbf{V}(\langle I, Z \rangle)$. This means that every set of points in k^n is the variety defined by some ideal containing the set Z. However, the only elements of $\bar{k}$ satisfying $x^2 + x = 0$ are 0 and 1, the elements of k, so $\overline{\mathbf{V}}(Z) = k^n$ and $\overline{\mathbf{V}}(\langle I, Z \rangle) = \mathbf{V}(I)$. Thus, if we restrict ourselves to ideals containing Z, we can still represent every subset of k^n and we can determine when two ideals represent the same set of points. As we will see in the next section, restricting our representations to ideals containing Z has some other advantages, as well.

3 Gröbner Bases

In this section, we sketch some of the theory of Gröbner bases. Although this theory has roots in the work of Macaulay as early as 1916, it really dates from Buchberger's thesis in 1965 [6]. There are now several good introductions to the subject; the reader seeking more details might consult the book by Cox, Little, and O'Shea [9] mentioned earlier or those by Becker and Weispfenning [4] and Adams and Loustaunau [1].

3.1 Motivation

To understand a little of the motivation for Gröbner bases, consider the problem of determining whether a given polynomial f belongs to an ideal $\langle f_1, \ldots, f_s \rangle$. If we work over a polynomial ring in one variable, the ideal is generated by a single polynomial, the greatest common divisor d of the set $\{f_1, \ldots, f_s\}$. There exist unique polynomials q and r with the degree of r strictly smaller than the degree of d and $f = qd + r$, and then f belongs to the ideal $\langle d \rangle$ if and only if the remainder r is 0. The polynomials d, q, and r are computed by standard algorithms.

For polynomials in more than one variable, the problem is more difficult. First, the ideal $\langle f_1, \ldots, f_s \rangle$ need not be generated by a single polynomial, so we must generalize our division algorithm to compute a remainder of f on division by the set $\{f_1, \ldots, f_s\}$. This is relatively straightforward, but it turns out that the remainder obtained this way is not uniquely determined. To get a unique remainder, which will be 0 if and only if $f \in \langle f_1, \ldots, f_s \rangle$, we need to use a special kind of generating set for the ideal. These generating sets are called Gröbner bases, and they provide the foundation for the algorithmic solution of many problems involving polynomials and ideals.

3.2 Definitions and basic properties

To define Gröbner bases, we need to specify an ordering on the set of monomials that satisfies certain conditions. It is somewhat more convenient to state things in terms of the n-tuples $(\alpha_1, \ldots, \alpha_n)$ rather than the monomials $x_1^{\alpha_1}, \ldots x_n^{\alpha_n}$, so let $\mathbb{N}^n$ be the set of n-tuples of nonnegative integers. There is an obvious isomorphism of semigroups between the set of monomials under the multiplication given in the previous section and $\mathbb{N}^n$ with component-wise addition.

Again, let k be an arbitrary field. A *monomial* or *term* order on $k[x_1, \ldots, x_n]$ is a relation $\succ$ on $\mathbb{N}^n$ (or equivalently on the set of monomials) satisfying the conditions that $\succ$ is a total order, $\succ$ is a well-ordering, and $\alpha \succ \beta$ implies $\alpha + \gamma \succ \beta + \gamma$ for all $\gamma \in \mathbb{N}^n$. The third condition is essentially a compatibility requirement between the order and the multiplication of monomials. We want to use the order to distinguish a leading, or highest, term in each polynomial. The third condition says that, if we multiply a polynomial by a monomial, the leading term of the result will be the product of the monomial and the leading term of the original polynomial.

Two commonly used monomial orders are the *lexicographic* order, in which $\alpha \succ \beta$ if and only if the leftmost nonzero entry in the difference $\alpha - \beta$ is positive, and the *graded reverse lexicographic* order, in which $\alpha \succ \beta$ if and only if $\sum_i \alpha_i > \sum_i \beta_i$ or $\sum_i \alpha_i = \sum_i \beta_i$ and the right-most nonzero entry in $\alpha - \beta$ is negative. Note, however, that each of these orders is defined only up to a permutation of the variables; there are really $n!$ versions of the lexicographic and graded reverse lexicographic orders. There are results indicating that, for many applications, the graded reverse lexicographic order is most efficient [3]. As we will see soon, some special orders, perhaps constructed from the graded reverse lexicographic, are also required for certain operations on ideals that are used in symbolic model checking.

We need some additional notation. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, we write x^α for the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$. Let $f = \sum_\alpha a_\alpha x^\alpha$ be a polynomial in $k[x_1, \ldots, x_n]$, and let $\succ$ be a monomial order. The *degree* of f, $\deg(f)$, is $\max\{\, \alpha \in \mathbb{N}^n \mid a_\alpha \neq 0 \,\}$. The *leading coefficient* of f, $\text{LC}(f)$,is $a_{\deg(f)}$. The *leading monomial* of f, $\text{LM}(f)$, is $x^{\deg(f)}$, and the *leading term* of f, $\text{LT}(f)$, is $\text{LC}(f) \cdot \text{LM}(f) = a_{\deg(f)} x^{\deg(f)}$.

Fix a monomial order. A finite subset $G = \{g_1, \ldots, g_t\}$ of an ideal I is a *Gröbner basis* for I (with respect to the given order) if and only if, for every $f \in I$, $\text{LT}(f)$ is divisible by one of the $\text{LT}(g_i)$. It is easy to see that every nonzero ideal has a Gröbner basis, and that any Gröbner basis for an ideal is also a basis for the ideal.

Suppose $\succ$ is a fixed monomial order on $k[x_1, \ldots, x_n]$ and $F = \{f_1, \ldots, f_s\}$ is an ordered s-tuple of polynomials. Then we can generalize the division algorithm for polynomials in one variable to show that every $f \in k[x_1, \ldots, x_n]$ can be written as a sum of multiples of the f_i and a polynomial r that is either 0 or a sum of monomials not divisible by any of $\text{LT}(f_1), \ldots, \text{LT}(f_s)$. We say that r is a *remainder* of f on division by F. The polynomial r depends on the way that the set F is indexed.

Buchberger gave an algorithm for constructing a Gröbner basis for a given ideal. The algorithm starts with a set of generators for the ideal. It then constructs an *S-polynomial* for a pair of elements of this set, and adds the remainder of the S-polynomial on division by the generating set to the set. It continues in this fashion until all the remainders are 0; at this point, the set of generators is a Gröbner basis. Various improvements in efficiency can be made by carefully choosing which S-polynomials to compute at a particular stage [7].

If G is actually a Gröbner basis for an ideal I and $f \in k[x_1, \ldots, x_n]$, then the remainder r of f on division by G is uniquely determined (i.e., does not depend on the order in which the elements of the basis are listed), and $f \in I$ if and only if $r = 0$. Buchberger's Gröbner basis algorithm thus yields an algorithm for determining whether a polynomial belongs to a given ideal. As noted in the previous section, we can also use

Gröbner bases to determine whether a polynomial is in the radical of a given ideal.

We say that a Gröbner basis G is *reduced* if the leading coefficients of the elements of G are all 1 and no monomial of an element of G lies in the ideal generated by the leading terms of the other elements of G. The key result is that, for a fixed monomial order, a nonzero ideal has a unique reduced Gröbner basis. The algorithm for finding a Gröbner basis can easily be extended to output this reduced Gröbner basis. Thus, we have an algorithm for determining whether two ideals $\langle f_1, \ldots, f_s \rangle$ and $\langle h_1, \ldots, h_t \rangle$ are equal.

3.3 Projections

Suppose that a concurrent system can be described in terms of n Boolean state variables, and let $\boldsymbol{F}$ be the field of two elements. We then represent the possible states of the system by the elements of the vector space $\boldsymbol{F}^n$. The transition relation of the system can then be regarded in the usual fashion as a subset T of $\boldsymbol{F}^{2n}$, where a point $(b_1, \ldots, b_n, b'_1, \ldots, b'_n) \in T$ if and only if there is a transition from the state represented by $(b_1, \ldots, b_n)$ to the one represented by $(b'_1, \ldots, b'_n)$. Suppose we have a set of points $C \subseteq \boldsymbol{F}^n$ corresponding to a formula ϕ. For symbolic model checking, we need to be able to describe the points corresponding to, for instance, the formula $EX\phi$. These are the points $(b_1, \ldots, b_n) \in \boldsymbol{F}^n$ such that there exists a point $(b'_1, \ldots, b'_n) \in C$ with $(b_1, \ldots, b_n, b'_1, \ldots, b'_n) \in T$. In the framework of algebraic geometry, this amounts to finding the projection of a subset of $\boldsymbol{F}^{2n}$ onto the first n coordinates. We can use Gröbner bases, with suitable monomial orders, to accomplish this.

Let R be the polynomial ring $\boldsymbol{F}[x_1, \ldots, x_n, x'_1, \ldots, x'_n]$ in $2n$ variables. We regard R as a ring of Boolean functions on $\boldsymbol{F}^{2n}$, as usual. Let $I = \langle f_1, \ldots, f_s \rangle$, and assume that the set Z consisting of the polynomials of the form $x_i^2 + x_i$ and $(x'_i)^2 + x'_i$ is contained in $\{f_1, \ldots, f_s\}$. (Recall that adding Z to the generating set of I does not change $\mathbf{V}(I)$.) Let R_1 be the subring consisting of polynomials in the variables $x_1, \ldots, x_n$ and let I_1 be the ideal $I \cap R_1$ of the ring R_1. We can show that any $(b_1, \ldots, b_n) \in V(I_1)$ extends to an element $(b_1, \ldots, b_n, b'_1, \ldots, b'_n) \in \mathbf{V}(I)$. In particular, if we take I to be an ideal with variety $\{\, (b_1, \ldots, b_n, b'_1, \ldots, b'_n) \in T \mid (b'_1, \ldots, b'_n) \in C \,\}$, then $\mathbf{V}(I_1)$ is the projection of this set on the first n coordinates. It is this projection that we need for model checking.

So the problem is to find I_1. Let $\succ$ be a monomial order satisfying the property that any monomial involving one of the x'_i is greater than any monomial involving only $x_1, \ldots, x_n$, and let $G = \{g_1, \ldots, g_s\}$ be a Gröbner basis of I with respect to $\succ$. If I contains Z, it can be shown that $G \cap R_1$ is a Gröbner basis for I_1. So we can find a Gröbner basis for I_1 as long as we can produce a suitable monomial order, and we can do that by, for example, modifying the graded reverse lexicographic order.

3.4 Complexity

It is natural to measure the size of a finite set F of polynomials in terms of the number of variables, the number of polynomials in F, the maximum degree of the polynomials, and the size of their coefficients. Given F, we are interested in these measures for a Gröbner basis for $\langle F \rangle$, as well as for the intermediate sets constructed in finding a Gröbner basis. In the general case, all of these measures behave fairly badly. For instance, examples are known where the construction of a Gröbner basis for an ideal generated by polynomials of degree less than or equal to d can involve polynomials of degree 2^{2^d} [12]. Over the field of two elements, however, all the coefficients are 0 or 1, and when our ideal includes all the $x_i^2 + x_i$, the only polynomials we have to consider are those in which no variable appears with degree greater than 1. I am not aware of specific complexity results for this case. Of course, just as with OBDDs, there are too many Boolean sets for all of them to have small representations in terms of Gröbner bases, so the interesting question is really one of characterizing the Boolean sets that do have such nice representations and understanding when the model checking process involves only such sets.

It is worth noting that there has been work on dynamic modification of the monomial order as the Gröbner basis calculation proceeds [10].

4 An Example

In this section we show how the machinery described in the preceding sections can be applied to verify a property of a small system. Consider the SMV code shown in Figure 1 (the numbers on the left in the module `prc` are inserted for reference, and are not part of the SMV program). This is the "mutex1" example distributed with SMV, with the fairness declarations deleted for simplicity. This system implements a mutual exclusion protocol.

We begin by describing the state variables. We can use one state variable for `turn` and two state variables for each of `s0` and `s1` to describe the state of the system, so we need 11 state variables for the transition relation (five for the current state, five for the next state, and one to keep track of which process is currently running, as required by the semantics of SMV). Figure 2 shows how we partition the variables. We encode the enumerated variables `s0` and `s1` by setting the corresponding pair of bits to $(0, 0)$ for `noncritical`, to $(0, 1)$ for `trying`, and to $(1, 0)$ for `critical`.

The next step is to find an ideal J such that $\mathbf{V}(J)$ is the transition relation, T. We have to capture the assignments made by the processes `pr0` and `pr1`. Our approach is to find polynomials whose zeros correspond to pairs of states in which the appropriate assignments are made.

Consider first `pr0`. Line (1) tells us that, if the system is in a state where `pr0` is running (i.e., when $x_6 = 0$), and `s0` is `noncritical` (i.e., when $(x_2, x_3) = (0, 0)$), the value of `s0` in the next state will be `noncritical` or `trying` (i.e., $(x_2', x_3') = (0, 0)$ or $(x_2', x_3') = (0, 1)$). So we need to find a set of polynomials whose common zeros are the points $(x_1, \ldots, x_5, x_1', \ldots, x_5', x_6)$ with $x_6 = 0$, $x_2 = 0$, $x_3 = 0$, $x_2' = 0$, and $x_3' = 0$ or 1. Since the condition on x_3' holds at all points, we can use the set $\{\, x_6, x_2, x_3, x_2' \,\}$. For calculations, it seems somewhat more convenient to take the single polynomial

$$f_1 = (x_6 + 1)(x_2 + 1)(x_3 + 1)(x_2' + 1) + 1,$$

which has the same zeros.

```
  MODULE main

  VAR
  s0: {noncritical, trying, critical};
  s1: {noncritical, trying, critical};
  turn: boolean;
  pr0: process prc(s0, s1, turn, 0);
  pr1: process prc(s1, s0, turn, 1);

  ASSIGN
  init(turn) := 0;

  SPEC
  EF((s0 = critical) & (s1 = critical))


  MODULE prc(state0, state1, turn, turn0)

  ASSIGN
  init(state0) := noncritical;
  next(state0) :=
    case
(1)    (state0 = noncritical) : {trying,noncritical};
(2)    (state0 = trying) & (state1 = noncritical): critical;
(3)    (state0 = trying) & (state1 = trying) & (turn = turn0): critical;
(4)    (state0 = critical) : {critical,noncritical};
(5)    1: state0;
    esac;
  next(turn) :=
    case
(6)    turn = turn0 & state0 = critical: !turn;
(7)    1: turn;
    esac;
```

Fig. 1. SMV program for mutual exclusion protocol

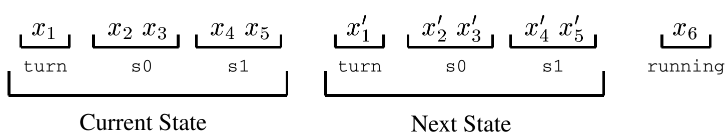

Fig. 2. State variables for transition relation

In a similar fashion, lines (2)–(4) yield polynomials

$$
\begin{aligned}
f_2 &= (x_6+1)(x_2+1)x_3(x_4+1)(x_5+1)x'_2(x'_3+1)+1\\
f_3 &= (x_6+1)(x_2+1)x_3(x_4+1)x_5(x_1+1)x'_2(x'_3+1)+1\\
f_4 &= (x_6+1)x_2(x_3+1)(x'_3+1)+1.
\end{aligned}
$$

Line (5) must be treated a little differently. It asserts that, if `pr0` is running and none of the first four guards in the case statement is true, then `next(s0)` = `s0`. There are two ways all the guards could fail: `s0` = `s1` = `trying` but `turn` = 1, and

$\texttt{s0 = trying}$ while $\texttt{s1 = critical}$. We will represent each of these conditions with a polynomial:

$$f_{5a} = (x_6+1)(x_2+1)x_3(x_4+1)x_5x_1(x_2'+1)x_3'+1$$
$$f_{5b} = (x_6+1)(x_2+1)x_3x_4(x_5+1)(x_2'+1)x_3'+1.$$

We note that it would also be possible to represent the negation of the guards on lines (1)–(4) directly, rather than explicitly listing the remaining cases. This approach is illustrated in the treatment of line (7) below.

Lines (6) and (7) describe the possible values of `next(turn)` while `pr0` is running. From line (6), we have

$$f_6 = (x_6+1)(x_1+1)x_2(x_3+1)x_1'+1,$$

using the fact that, for `pr0`, $\texttt{turn0} = 0$.

Line (7) tells us that, while `pr0` is running, `turn` does not change unless the guard of line (6) is satisfied. We want a polynomial that is zero at exactly the points where $x_6 = 0$, the guard of line (6) is false (so $(x_1+1)x_2(x_3+1) = 0$), and $x_1 = x_1'$. A polynomial that is zero at exactly these points is

$$f_7 = (x_6+1)\left((x_1+1)x_2(x_3+1)+1\right)(x_1+x_1'+1)+1.$$

The variable `s1` is not assigned while `pr0` is running. The semantics of SMV then imply that $\texttt{next(s1)} = \texttt{s1}$ if `pr0` is running. We can express this condition with the polynomial

$$f_8 = (x_6+1)(x_4+x_4'+1)(x_5+x_5'+1)+1.$$

The points $(x_1,\ldots,x_5,x_1',\ldots,x_5',x_6) \in T$ corresponding to pairs of states in which `pr0` is running in the current state are those where one of $f_1 \ldots f_{5b}$ is zero, one of f_6 or f_7 is zero, and f_8 is zero. Since a product of polynomials is zero if and only if at least one of the factors is zero, these are the points where the three polynomials $f_1f_2f_3f_4f_{5a}f_{5b}$, f_6f_7, and f_8 are all zero. In other words, the points in the transition relation with $x_6 = 0$ form the variety of the ideal $I_{\texttt{pr0}} = \langle f_1f_2f_3f_4f_{5a}f_{5b}, f_6f_7, f_8\rangle$.

In a similar fashion, we construct an ideal $I_{\texttt{pr1}}$ whose variety is the set of points in T with $x_6 = 1$. If we set $I = I_{\texttt{pr0}} \cdot I_{\texttt{pr1}}$ and $J = \langle I, Z\rangle$, where $Z = \{x_1^2+x_1,\ldots,x_5^2+x_5,(x_1')^2+x_1',\ldots,(x_5')^2+x_5',x_6^2+x_6\}$, then $T = \mathbf{V}(J)$.

The property we want to check is $EF(\texttt{s0 = critical} \wedge \texttt{s1 = critical})$. Let $\phi = (\texttt{s0 = critical} \wedge \texttt{s1 = critical})$. So we want to find the least fixed point of $\tau = \lambda y.\phi \vee EXy$. Given a description of y as a variety, we need to express the points corresponding to $\phi \vee EXy$ as the variety of some ideal. To do this, we need to describe the points satisfying ϕ as a variety, and we need to compute the ideal defining the variety EXy.

The points $(x_1,\ldots,x_5,x_1',\ldots,x_5',x_6)$ for which ϕ holds are those corresponding to system states in which both `s0` and `s1` are `critical`, i.e., those in which $x_2 = x_4 = 1$ and $x_3 = x_5 = 0$. These are the points in the variety of the ideal $I_\phi = \langle x_2(x_3+1)+1, x_4(x_5+1)+1\rangle$.

To find the ideal corresponding to EXy, we first need to specify that the polynomials defining y are zero in the next state. In our setting, this is accomplished by applying a homomorphism of rings that replaces the x_i by the corresponding x_i'. Let $R = \boldsymbol{F}[x_1, \ldots, x_5, x_1', \ldots, x_5', x_6]$ and let $\nu\colon R \to R$ be the (k-linear) ring homomorphism mapping each x_i to x_i', for $i = 1, \ldots, 5$, each x_i' to 0, and x_6 to x_6. If $f \in R_1 = \boldsymbol{F}[x_1, \ldots, x_6]$ is a polynomial in the x_i, $\nu(f)$ is the corresponding polynomial in the variables $x_1', \ldots, x_5', x_6$.

Then if y corresponds to the variety $\mathbf{V}(h_1, \ldots, h_s)$, the variety corresponding to EXy is the projection onto the first n coordinates of the variety of the ideal $I_{y'} = \langle T, \nu(h_1), \ldots, \nu(h_s), Z\rangle$. We find the ideal defining this variety using the methods discussed in Section 3.3: We construct a Gröbner basis $G_{y'}$ for $I_{y'}$ with respect to a suitable order, and take the elements of $G_{y'}$ that lie in the subring R_1. If $G_1 = R_1 \cap G_{y'}$, then the variety defined by $\langle G_1\rangle \cdot I_\phi$ corresponds to the points satisfying the formula $\phi \vee EXy$. In this fashion, we can find the least fixed point of $\lambda y.\phi \vee EXy$.

I used the program *Macaulay* [2] to carry out these calculations. *Macaulay* provides facilities for defining rings, ideals, and homomorphisms, and for carrying out a variety of Gröbner basis calculations. Many of these calculations could have been done using other computer algebra systems; *Macaulay* seemed to be the most convenient for these experiments.

The Gröbner basis found by *Macaulay* for the ideal I_μ whose variety is the least fixed point of $\lambda y.\phi \vee EXy$ consists of the six polynomials $x_1^2 + x_1$, $x_2 + 1$, x_3, $x_4 + 1$, x_5, and $x_6^2 + x_6$. (Note that the first and last of these are zero at all points of $\boldsymbol{F}^n$.) The variety $\mathbf{V}(I_\mu)$ consists of the points $(x_1, \ldots, x_5, x_1', \ldots, x_5', x_6)$ where $x_2 = 1, x_3 = 0$, $x_4 = 1$, and $x_5 = 0$. These are the points where `s0` and `s1` are both `critical`; this tells us that it is not possible to reach a state where both `s0` and `s1` are `critical` (i.e., where ϕ holds) from a state where at least one is not `critical`. In particular, no state where both `s0` and `s1` are `critical` is reachable from the initial state, since the initial conditions specify that `s0` and `s1` are `noncritical`. We can verify this by expressing the initial conditions as the zeros of an ideal, say $I_{\texttt{init}} = \langle x_2, x_3, x_4, x_5, x_6\rangle$, and computing the ideal of the intersection of the varieties $\mathbf{V}(I_\mu)$ and $\mathbf{V}(I_{\texttt{init}})$. *Macaulay* reports that the constant polynomial 1, which has no zeros, is a Gröbner basis for this ideal, and we see that the intersection is empty. We conclude that $EF\phi$ is false in the initial state.

Alternatively, we could have found the set of reachable states by starting from $I_{\texttt{init}}$, and taken the intersection with this variety at each stage. (This corresponds to running SMV with the `-f` flag.)

Macaulay runs as an interpreter that can be used interactively or can execute scripts. A script to check the property $EF($`s0 = critical` $\wedge$ `s1 = critical`$)$ took about 10 seconds to execute on a PC with a 100 MHz Pentium and 16 MB of memory, running Linux. *Macaulay* allocated 755 KB of memory in the course of this calculation. For comparison, on the same machine SMV took approximately 0.1 seconds to check the same property, and allocated just over 917 KB. SMV, of course, was building OBDDs from the code shown in Figure 1, while for *Macaulay*, I had manually translated this code into the polynomials described above.

5 Discussion

In this paper, I have shown how techniques from computational algebraic geometry can be used for symbolic model checking. This approach may provide a useful supplement to existing methods based on OBDDs, and may also provide important theoretical insights by allowing the application of deep results in algebraic geometry to the model checking problem. Additional research will be needed to determine whether these potential advantages are borne out.

Macaulay, the program I used for the calculations described in the previous section, was intended for use in a much more general setting. It supports, for instance, calculations over fields of characteristic up to about 32,000, rather than just characteristic 2. Its data structures and algorithms are therefore not optimized for the cases used in symbolic model checking. Furthermore, it runs as an interpreter. For that reason, the difference in execution time between *Macaulay* and SMV does not seem to carry much significance for assessing the practicality of these methods. Although some further investigation of the practicality of symbolic model checking using the techniques from algebraic geometry can probably be done using tools like *Macaulay*, more serious study will likely require building a prototype tool designed specifically for that purpose. Examples like the one in the previous section suggest that it should be fairly easy to build a tool that would work directly from specifications given in the SMV input language.

There are several directions in which the framework proposed here might be generalized. For instance, in the example of Section 4, I worked with polynomials over the field of two elements. This has some clear advantages and seems to be the most natural analog of the OBDD approach. Working over the field of order 2^k, however, might allow much more efficient encoding of conditions involving k-bit blocks of state variables. Similarly, working over fields of characteristic greater than 2 would correspond to some of the non-binary generalizations of OBDDs.

It is difficult to predict exactly what theorems of algebraic geometry might be applicable to symbolic model checking, but some general directions can be sketched. For instance, there is a rich collection of invariants of varieties and ideals, including such things as notions of dimension and degree. Many of these invariants are likely to be related to the difficulty of carrying out symbolic model checking. Algebraic geometry also provides good machinery for handling such things as the action of groups on varieties, maps between varieties, and the properties of intersections of varieties. It might therefore provide new ways to understand and take advantage of symmetries of the system being checked, abstraction to simpler systems, or the effects of constraints representing the interface between a subsystem and its environment. Results in these directions might give information about, for instance, the kinds of Boolean sets arising in fixed point calculations and thus even have implications for model checking using OBDDs.

Acknowledgments

This research was partially supported by the National Science Foundation under Grant No. CCR-9407182. I am grateful to David Cox for helpful discussions about Gröbner bases, to Jay Corbett for clarifying many of the details of SMV, for making a lightly loaded SparcStation available for some of my experimentation and for helpful comments on earlier drafts of this paper, and to Nicholas Schmitt for providing me with several versions of his Gröbner basis package, *Ideal*, which I used in my initial exploration of these ideas.

References

1. W. W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases*, volume 3 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1994.
2. D. Bayer and M. Stillman. *Macaulay: A System for Computation in Algebraic Geometry and Commutative Algebra*. Source and object code available for Unix and Macintosh computers. Contact the authors, or download from `math.harvard.edu` via anonymous ftp., 1982–1994.
3. D. Bayer and M. Stillman. A criterion for detecting m-regularity. *Invent. Math.*, 87:1–11, 1987.
4. T. Becker and V. Weispfenning. *Gröbner Bases*, volume 141 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1993.
5. R. E. Bryant. On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. *IEEE Transactions on Computers*, 40(2):205–213, Feb. 1991.
6. B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.
7. B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232. D. Reidel, 1985.
8. J. Burch, E. Clarke, K. McMillan, D. Dill, and L. Hwang. Symbolic model checking: 10^{20} states and beyond. In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 428–439, 1990.
9. D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1992.
10. P. Gritzmann and B. Sturmfels. Minkowski addition of polytopes: Computational complexity and applications to Gröbner bases. *SIAM Journal on Discrete Mathematics*, 6(2):246–269, 1993.
11. R. Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1977.
12. E. Mayr and A. Meyer. The complexity of the word problem for commutative semigroups and polynomial ideals. *Adv. in Math.*, 1982.
13. K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, Boston, 1993.