Computers Math. Applic. Vol. 22, No. 12, pp. 61–66, 1991
Printed in Great Britain. All rights reserved

0097-4943/91 $3.00 + 0.00
Copyright© 1991 Pergamon Press plc

A DYNAMIC PROGRAMMING METHOD FOR BUILDING FREE ALGEBRAS

IRVIN ROY HENTZEL
Department of Mathematics, Iowa State University
Ames, Iowa 50011, U.S.A.
DAVID POKRASS JACOBS
Dept. of Computer Science, Clemson University
Clemson, South Carolina 29634-1906, U.S.A.

(Received March 1991)

Abstract—We are interested in deciding if a given nonassociative polynomial h is an identity for a variety of nonassociative algebras. We present an algorithm for constructing a certain homomorphic image of a free nonassociative algebra which is sufficient to answer the question. The algorithm resembles dynamic programming in that the algebra is built by constructing a sequence of subspaces; the basis of each subspace is determined by the basis of previous subspaces. The number of arithmetic operations required to construct the algebra is bounded by a polynomial in the degree of h and the dimension of the resulting algebra.

INTRODUCTION

This paper studies an algorithm for a certain algebraic problem. In particular, we are interested in determining if a given nonassociative polynomial is an identity for a class of nonassociative algebras. Throughout this paper Γ will be a field. A *nonassociative algebra over* Γ is a vector space (over Γ) equipped with a binary operation, known as multiplication, that is linear in both arguments:

$$x(y+z) = xy + xz$$
$$(x+y)z = xz + yz$$
$$\alpha(xy) = (\alpha x)y = x(\alpha y), \quad \alpha \in \Gamma.$$

Given a set I of nonassociative polynomials, such as

$$xy - yx, \tag{1}$$

$$x^2x^2 - (x^2x)x, \tag{2}$$

a nonassociative algebra A *satisfies* the polynomials if they are identically zero on A. In this case we say the polynomials are *identities.* (We sometimes informally omit the adjective "nonassociative" when speaking of algebras and polynomials). The *variety* of algebras defined by I is the class of all algebras over Γ that satisfy each member of I. Polynomials (1) and (2) define the variety of commutative, fourth-power-associative algebras. The general problem motivating this paper is: *given a finite set $I = \{f_i\}$ of defining identities, and given a polynomial h, is h an identity for all members belonging to the variety defined by I?*

Given a monomial αw, $\alpha \in \Gamma$, and given an indeterminate x, the *degree of* αw *in* x, denoted

$$\deg_x(\alpha w),$$

The preparation of this paper was supported in part by NSF Grant CCR8905534.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

is the number of times x occurs in the word w. The *total degree* of a monomial is the sum of the degrees over all of its indeterminates, and the degree of a nonassociative polynomial f, denoted $\deg(f)$, is the largest total degree of its monomials. Two monomials are said to have the same *type* if they have the same degree in each indeterminate x. A nonassociative polynomial is ***homogeneous*** if all of its monomials have the same type. Let f be a polynomial, and suppose Γ contains more elements than the degree of each monomial in each indeterminate. Then in any algebra over Γ, f is an identity if and only if each of its homogeneous components are identities [1]. A homogeneous polynomial having degree 1 in each of its indeterminates is called *multilinear.* Polynomials (1) and (2) are each homogeneous, but only (1) is multilinear. Any homogeneous polynomial f can be transformed to a multilinear polynomial f' through a process known as *linearization.* If Γ either has characteristic 0 or has characteristic greater than k, and I is a set of homogeneous polynomials whose degree in each indeterminate is at most k, then the variety $\mathbf{V}$ defined by I is the same as the variety defined by the set of complete linearizations of the identities of I [2]. The linearized form of (2) is

$$\sum_{\pi \in S_4} (x_{\pi(1)}x_{\pi(2)})(x_{\pi(3)}x_{\pi(4)}) - ((x_{\pi(1)}x_{\pi(2)})x_{\pi(3)})x_{\pi(4)}. \tag{2a}$$

By the above comments, if we are willing to make the assumption that the characteristic of Γ is either zero or exceeds the degrees of each monomial in each indeterminate, we can replace each nonhomogeneous member of I with its homogeneous components. We may then linearize each polynomial. We assume, therefore, that I contains only multilinear identities. We also may assume that each member of I has degree at least two, since any degree one identity defines a trivial variety. The polynomial h is assumed to be homogeneous but not necessarily multilinear.

To add concreteness, consider a typical instance of this decision problem. First, a *Jordan* algebra is a commutative algebra satisfying the identity (a^2, b, a), where the associator (a, b, c) denotes $(ab)c - a(bc)$. All Jordan algebras satisfy (1) and (2), and so it is reasonable to wonder how close algebras satisfying (1) and (2) are to being Jordan. For example, one might ask if the *square*

$$h = (a^2, b, a)^2, \tag{3}$$

is an identity in a variety $\mathbf{V}$ defined by (1) and (2).

In this paper, we approach the general problem by considering the *free algebra* $F = F[X]$, consisting of all polynomials over X, where X is the set of indeterminates appearing in h. Let $I(F)$ be the ideal in F generated by all polynomials obtained by making substitutions with arbitrary members from F for the indeterminates of each $f \in I$. The algebra $F_I = F/I(F)$ is known as the *free algebra in* $\mathbf{V}$ *on* X. The nonassociative polynomial h is an identity in $\mathbf{V}$ if and only if h is an identity in the algebra F_I. Thus, the question involving an identity for all of $\mathbf{V}$ is reduced to a question involving only a particular algebra. For more background , see [1]. For the remainder of this paper, the symbols Γ, $I = \{f_i\}$, $\mathbf{V}$, X, and h will each have the meanings given above.

DYNAMIC PROGRAMMING METHOD

Dynamic programming is a general term describing a class of algorithms that solve the problem at hand by inductively solving a series of smaller problems of the same type. At each stage of the algorithm, results are saved to be used in subsequent stages.

Any finite dimensional algebra of dimension d can be described by specifying a basis $\{b_1, \ldots, b_d\}$ and a collection $\{\delta_{ijm}\}$ of d^3 *structure constants* such that for all i, j, $1 \leq i, j \leq d$,

$$b_i b_j = \sum_{m=1}^{d} \delta_{ijm} b_m.$$

This specifies the rules for multiplying two basis elements, and multiplications involving arbitrary linear combinations of basis elements can be defined by extending these rules "linearly". When an algebra is described in this way, it is straightforward to check if it satisfies a multilinear polynomial; if the polynomial is zero for all substitutions of basis elements, then the polynomial is identically zero on the algebra. This fact is crucial if we wish to construct an algebra that satisfies a finite set I of multilinear identities; at each step in our construction we ensure that the basis members satisfy the polynomials in I. In general, it is impossible for us to write down a basis with structure constants for the entire algebra F_I since it may be infinite dimensional, but we *can* construct a certain finite dimensional homomorphic image of it.

Let $X = \{x_1, \dots, x_t\}$ be the set of indeterminates appearing in h, the homogeneous polynomial in question, and let n_i be the degree of h in x_i. Then

$$n = \sum_{i=1}^{t} n_i$$

is the degree of h. For $i = 1, \dots n$, let K_i be the subspace of F spanned by all degree i words w over X where $\deg(w) = i$, and for each j, $0 \leq \deg_{x_j}(w) \leq n_j$. Next, define

$$K = K_1 + K_2 + \dots + K_n.$$

Let L be the subspace of F spanned by words w, where for some j, $\deg_{x_j}(w) > n_j$. L is an ideal of F, and $F = K + L$. In our example involving (3), K consists of linear combinations of all words over $\{a, b\}$ of degree at most 8, with at most 6 a's and at most 2 b's.

We now present an algorithm for constructing a certain algebra A. Our algorithm was motivated by Kleinfeld's construction, by hand, of a 107–dimensional alternative algebra [3]. The algebra is constructed by determining a basis and table of structure constants. Its basis is selected from the words in K. The algebra is constructed in n stages. *The key idea is that at stage i, a set B_i of degree i basis members are selected from K_i, and structure constants are determined for each pair of basis elements b', b'' that were selected at stages i', i'' respectively, for which $i = i' + i''$.*

Initially, at stage 1 the basis B_1 is defined to be X. At the beginning of stage i, $i \geq 2$, we let

$$C_i = \{(b_u)(b_v) \in K_i | b_u, b_v \text{ are basis}\}.$$

At this point all members of C_i are potential basis members to be included at stage i. During stage i, the basis members from C_i are chosen by first constructing a certain set of polynomials: For each defining identity $f(y_1, \dots, y_r) \in I$, consider each sequence $b_1, \dots, b_r$ of basis words, obtained at earlier stages, for which

$$\sum_{t=1}^{r} \deg(b_t) = i \quad \text{and} \quad \sum_{t=1}^{r} \deg_{x_j}(b_t) \leq n_j \quad \text{for each } j.$$

Now, expand the equation $f(b_1, \dots, b_r) = 0$ by using the previously determined portion of the table. Before expansion, the expression $f(b_1, \dots, b_r)$ is a sum of terms each containing a scalar times a product involving $b_1, \dots, b_r$ in some association and in some permutation. To expand $f(b_1 \dots b_r)$, the table is applied until the expression has been rewritten into a linear combination of the form

$$\sum_{u,v} \alpha_{u,v}(b_u)(b_v), \quad \deg(b_u) + \deg(b_v) = i, \tag{4}$$

where b_u and b_v are basis words. Of course, at this point no further expansion is possible because the table is not completed for degree i. By setting (4) equal to zero we obtain a relation among the members of C_i.

These equations are linear over the set C_i and impose the identities in I among lower degree basis elements. Since the system of equations is homogeneous, a solution must exist. The degree i basis, B_i, is found by selecting a basis from C_i by using ordinary linear algebra to solve the equations. The left side of each equation is treated as a linear combination over all of C_i. Therefore, if a particular word $w \in C_i$ appears with a zero coefficient in every equation, it will necessarily become a basis member. Solving the equations provides both the new basis members, and structure constants for expressing products $b_u \cdot b_v$ in terms of the basis, where $(b_u)(b_v) \in C_i$. Note that only those polynomials $f \in I$ having degree at most i are used at stage i. Hence the early stages of the algorithm will just select all words from C_i if no members of I can impose relations upon them. The algorithm is sketched in Figure 1. For two words $w_1, w_2 \in F$, we denote their free product in F by $(w_1)(w_2)$. However if these words happen to become basis elements in the algebra under construction, we denote their product in A by $w_1 \cdot w_2$. More generally, we will denote the product of two elements g_1 and g_2 in A with $g_1 \cdot g_2$.

```
Initialize $B_1$ with X.
for i := 2 to n do begin
    for each $f(y_1, \cdots, y_r) \in$ I of degree $\le$ i do
        for each sequence of basis members $b_1, \cdots, b_r$
            if $\sum \deg(b_k) = i$ and $f(b_1, \cdots, b_r) \in K_i$ then
                expand $f(b_1, \cdots, b_r)$ into terms of form (4).
    Solve equations, getting $B_i$ from $C_i$ and dependence relations R.
    Complete the table for degree i:
        for each pair $b_1, b_2$ of basis members with $\deg(b_1) + \deg(b_2) = i$
            if $(b_1)(b_2) \notin K$ then
                define $b_1 \cdot b_2 = 0$
            else if $(b_1)(b_2)$ is a basis word in R then
                define $b_1 \cdot b_2 = (b_1)(b_2)$
            else
                define $b_1 \cdot b_2 = \sum_{b \in B_i} \alpha_b b$ according to R.
end {main loop}
Complete the table: set $u \cdot v = 0$ if $\deg(u) + \deg(v) > n$.
```

Figure 1.

THEOREM 1. *The algebra A constructed in Figure 1 is isomorphic to $F/(I(F) + L)$.*

PROOF. Note that A has a basis whose elements are words in $K \subseteq F$. Since this basis contains $B_1 = X$, we may consider the identity map $x_i \rightarrow x_i$ from X into A. Because F is the free algebra on X, this map may be extended uniquely to a homomorphism [1]

$$\psi : F \rightarrow A.$$

We first note that since every basis element is a product, in A, of the elements in X, X generates the algebra A. Therefore this homomorphism is *onto* A, and so $A \cong F/J$ where J is the kernel of ψ. We will show $J = I(F) + L$.

Next, let w be a word in L. We claim $\psi(w) = 0$. The degree of w must be at least 2, and so we may write $w = (w_1)(w_2)$. Since $\psi(w) = \psi(w_1) \cdot \psi(w_2)$, by an induction argument on the degree of w, we may assume $w_1 \notin L$ and $w_2 \notin L$. Thus $w_1, w_2 \in K$. Any element $\psi(w) \in A$ is the vector obtained by multiplying out the generators in w according to the table for A. Moreover, since the members of I are homogeneous (in fact, multilinear), by our construction $\psi(w)$ is a linear combination $\sum \alpha_t b_t$ of basis words in A, where any basis word b_t having nonzero coefficient α_t must have the same type as w. Therefore $\psi(w_1) \cdot \psi(w_2)$ can be written as a linear combination of elements of the form $b_u \cdot b_v$, in which b_u and b_v are basis members in A, and where w_1 and b_u are of the same type, and w_2 and b_v are of the same type. Therefore the words

$(b_u)(b_v) \in L$. If $\deg(b_u) + \deg(b_v) > n$ then $b_u \cdot b_v = 0$ by step 17. We may assume then that $\deg(b_u) + \deg(b_v) = i = \deg(w) \leq n$. At stage i the products $b_u \cdot b_v$ are determined. But by lines 10 and 11 these products are defined to be 0. Hence $\psi(w) = \psi(w_1) \cdot \psi(w_2) = \sum \alpha_{uv} b_u \cdot b_v = 0$, and so $L \subseteq J$.

Now let $f \in I$. We claim that A satisfies f. Since f is multilinear it suffices to show that $f(b_1, \ldots b_r) = 0$ for any choice of basis elements b_j. If $f(b_1, \ldots b_r) \in K_i$ for some i then at stage i of the algorithm we ensure that $f(b_1, \ldots, b_r) = 0$. If $f(b_1, \ldots, b_r) \in L$ by the preceding argument this must be zero. It follows that A satisfies each member of I. Therefore we must have $I(F) \subseteq J$. We now have $I(F) + L \subseteq J$.

To complete the proof, it is sufficient to show that $\dim(F/(I(F) + L) \leq \dim(A)$. It is therefore sufficient to show that $F/(I(F) + L)$ is spanned by a set whose cardinality is at most $\dim(A)$. Let $\{b_j\}$ be the basis obtained in Figure 1 for A. Let $\overline{x}$ denote the image of x under the natural homomorphism from F to $F/(I(F) + L)$. We claim the co-sets $\overline{b_j}$ span $F/(I(F) + L)$. Let w be a word. It suffices to show $\overline{w}$ can be written in terms of the $\overline{b_j}$. We may assume $w \in K$, for otherwise $\overline{w} = 0$. Since $X \subseteq \{b_j\}$ we may also assume $\deg(w) \geq 2$ and write $w = (w_1)(w_2)$. By induction we may assume $\overline{w_1} = \sum \alpha_{1j} \overline{b_j}$ and $\overline{w_2} = \sum \alpha_{2j} \overline{b_j}$, and hence $\overline{w} = \overline{(w_1)(w_2)} = (\overline{w_1})(\overline{w_2}) = \sum \beta_{kj} (\overline{b_k})(\overline{b_j}) = \sum \beta_{kj} \overline{(b_k)(b_j)}$. Let $i = \deg(w)$. Consider the word $(b_k)(b_j)$. At stage i, this becomes a basis member or we will have $(b_k)(b_j) = \sum \alpha_s b_s + f$ where $f \in I(F)$. Hence $\overline{(b_k)(b_j)} = \overline{\sum \alpha_s b_s + f} = \sum \alpha_s \overline{b_s}$ and we are done. This completes the proof. ∎

Corollary. *The polynomial h is an identity for $\mathbf{V}$ if and only if $\psi(h) = 0$.*

Proof. The identities for $\mathbf{V}$ are precisely $I(F)$. If h is an identity for $\mathbf{V}$ then

$$h \in I(F) \subseteq I(F) + L = \ker(\psi)$$

and so $\psi(h) = 0$. Conversely, if $\psi(h) = 0$ then by Theorem 1, $h \in \ker(\psi) = I(F) + L$. We must have $h = f + l$ where $f \in I(F)$ and $l \in L$. We can write $f = f' + f''$, where $f' \in K$ and $f'' \in L$. By the comments made in the introduction, both f' and f'' must be identities, and so in particular $f' \in I(F)$. We have $h = f' + (f'' + l)$. Since $h \in K$, it follows that $f'' + l = 0$, and so $h = f' \in I(F)$. This implies h is an identity for $\mathbf{V}$. ∎

This corollary tells us that to determine if h is an identity we merely expand h according to the rules of multiplication for A. We summarize some of the properties of the algebra A, the resulting basis, and structure constants:

i) A is generated by X.
ii) A's basis consists of X together with certain words on X.
iii) Given a basis word b, $\deg(b) \geq 2$, both left and right factors of b are basis words.
iv) Given a basis word b, $\deg(b) \geq 2, b$ is the product in A of its left and right factors.

Interestingly, Hall's basis for Lie rings [4] also shares these properties.

ANALYSIS OF THE ALGORITHM

The algorithm described in this paper has been implemented, and is the heart of a computer algebra system called *Albert*, described in [5]. Preliminary experiments have been very promising. For example, showing a certain degree 9 polynomial was not an identity took several months of computer time using an earlier technique [6], but Albert solved the same problem in about 79 hours. In this section we explain the improvement.

Theorem 2. *For each fixed finite I and fixed finite field Γ, the construction of the algebra A by the dynamic programming method takes time bounded by a polynomial in $\deg(h)$ and the dimension of A.*

Proof. Let $d = \dim(A)$ and $n = \deg(h)$. Since the main loop iterates n times, it suffices to show that each iteration takes time a polynomial in d. Let r be the largest degree over each $f \in I$. Recall that equations are generated by selecting basis elements and substituting them into arguments of the $f \in I$. Thus the number of equations generated during the construction is at most $|I| d^r$.

We claim each such equation can be expanded in time $O(d^3)$. To see this, first observe that two linear combinations over the basis can be multiplied, and the product written in terms of the basis, in time $O(d^3)$. For multiplying the two linear combinations involves multiplying d^2 monomials of the form $(\alpha_i b_i)(\beta_j b_j)$, and each such product can be found in time $O(d)$ by using the already generated portion of the multiplication table. Suppose f has degree r. We wish to expand $f(b_{j_1},\ldots,b_{j_r})$ into a linear combination of pairs of basis elements. This can be done by repeatedly applying the above operation within each monomial of f, and multiplying in the normal order imposed by the parenthesis. To do this, there are $r - 1$ such operations for each monomial. (In fact, the last operation only multiplies two linear combinations into a linear combination over basis pairs.) Since r and the number of monomials in f is constant, this takes time $O(d^3)$.

Each expanded equation appears in exactly one system of equations. Any system can be solved in time polynomial in its size. The number of rows is polynomial in d. The number of columns is at most d^2, since each column corresponds to a pair of basis elements. Because Γ is finite we are guaranteed that arithmetic operations take constant time. Therefore all systems of equations can be generated and solved in time polynomial in d. ∎

Besides dynamic programming method, there is another method for constructing a basis for $F/(I(F)+L)$, which we call the *brute force* method, and which we now briefly describe. Imagine a large matrix M whose columns are each indexed by the words of K. Construct a set of polynomials that span the subspace $K \cap I(F)$ as follows. Take $f \in I$ and suppose it has degree $r \leq n$. Consider all finite sequences $a_1, a_2, \ldots, a_t$ of words $a_i \in K$ such that

$$f(a_1,a_2,\ldots,a_r)S_{a_{r+1}}S_{a_{r+2}}\cdots S_{a_t} \in K.$$

Here S_{a_i} can be either a right or a left multiplication by a_i. We do this for each f. These generated identities form a spanning set for the vector space $I(F) \cap K$. Each such identity is then written as a linear combination over K, and entered as a row of coefficients in M. This matrix can be reduced to row canonical form, using ordinary linear algebra. The basis columns of M will correspond to a basis for $F/(I(F)+L)$.

REMARK. The brute force method does not run in time polynomial in deg (h) and dim (A).

PROOF. Let I be the set containing only the associative law. Let F be the free algebra on a. Let L_n be the ideal in F generated by a^{n+1}. Then $A_n = F/(I(F) + L_n)$ is the commutative associative algebra with basis a, $a^2, \ldots a^n$, and so the dimension of A_n is n. Constructing A_n using the brute force method requires at least

$$\sum_{i=1}^{n} \frac{1}{i}\binom{2i-2}{i-1}$$

arithmetic operations since this is the number of words in K. This is not polynomial in n. ∎

REFERENCES

1. K.A. Zhevlakov, A.M. Slin'ko, I.P. Shestakov and A.I. Shirshov, *Rings That Are Nearly Associative*, Academic Press, New York, (1982).
2. J.M. Osborn, Varieties of algebras, *Advances in Mathematics* **8**, 163–369 (1972).
3. E. Kleinfeld, On centers of alternative algebras, *Communications in Algebra* **8** (3), 289–297 (1980).
4. M. Hall, A basis for free lie rings and higher commutators in free groups, *Proc. Amer. Math. Soc.* **1**, 575–581 (1950).
5. D.P. Jacobs, S. Muddana and A.J. Offutt, A computer algebra system for nonassociative identities, Presented at the *Proceedings of the Fifth International Conference on Hadronic Mechanics and Nonpotential Interactions*, Cedar Falls, (1990) (to appear).
6. I.R. Hentzel and D.J. Pokrass, Verification of non-identities in algebras, *Proceedings of the 1988 International Symposium on Symbolic and Algebraic Computation, Lecture Notes in Computer Science*, Vol. 358, (Edited by P. Gianni), pp. 496–507, Springer-Verlag, (1989).