Nonassociative Structures on Polynomial Algebras Arising from Bio-operations on Formal Languages

An Application of Computer Algebra to Nonassociative Systems

Murray R. Bremner, Michael J. Hancock, Yunfeng Piao
CALC: Centre for Algebra, Logic and Computation
Department of Mathematics and Statistics, University of Saskatchewan
106 Wiggins Road (McLean Hall), Saskatoon, SK, S7N 5E6, Canada
bremner@math.usask.ca, mjh750@mail.usask.ca, yup509@mail.usask.ca

ABSTRACT

We consider sequential insertion and deletion, and contextual insertion and deletion, on the free monoid Σ^* where $\Sigma = \{x\}$; in each case the result can be regarded as either a set or a multiset. Over any coefficient field $\mathbb{F}$ the vector space with basis Σ^* is linearly isomorphic to the polynomial algebra $\mathbb{F}[x]$; each operation on Σ^* extends bilinearly to give a new algebra structure (not necessarily commutative or associative) on $\mathbb{F}[x]$. We determine the polynomial identities of degree ≤ 5 satisfied by these structures.

Categories and Subject Descriptors

F.2.1 [**Numerical Algorithms and Problems**]: Computations in finite fields; F.4.3 [**Formal Languages**]: Algebraic language theory; G.1.3 [**Numerical Linear Algebra**]: Sparse, structured, and very large systems; I.1.2 [**Algorithms**]: Algebraic algorithms; J.2 [**Physical Sciences and Engineering**]: Mathematics and statistics; J.3 [**Life and Medical Sciences**]: Biology and genetics

General Terms

Algorithms, Languages, Theory

Keywords

Computer algebra, linear systems, finite fields, nonassociative algebra, polynomial identities, formal languages, bio-operations, DNA computing

1. INTRODUCTION

In the theory of DNA computing, the processes of molecular genetics are expressed in terms of operations on formal languages; for a survey of this area, see Paŭn, Rozenberg and Salomaa [9]. In particular, many variations on insertion and deletion have been studied by theoretical computer scientists; see especially the doctoral thesis of Kari [7].

Given a finite nonempty set Σ, we write Σ^* for the free monoid generated by Σ. For any field $\mathbb{F}$, the vector space with basis Σ^* is linearly isomorphic to $\mathbb{F}[\Sigma]$, the free associative algebra over $\mathbb{F}$ generated by Σ. Bilinear extension of the natural monoid operation (concatenation) on Σ^* induces the natural associative algebra structure on $\mathbb{F}[\Sigma]$. Other operations on Σ^* induce other, usually nonassociative, algebra structures on $\mathbb{F}[\Sigma]$. These structures can be regarded as linearizations of the corresponding operations on Σ^*. This approach allows us to study properties of language operations which cannot be expressed in terms of monoids: in particular, the polynomial identities satisfied by the nonassociative structures. This provides a connection between operations on formal languages and varieties of nonassociative algebras; for earlier work in this area, see Bremner [1, 2].

In this paper we consider the simplest case $\Sigma = \{x\}$ and determine the polynomial identities of degree ≤ 5 satisfied by the nonassociative products on $\mathbb{F}[x]$ induced by set and multiset versions of sequential and contextual insertion and deletion. Our methods depend heavily on computer algebra, and in particular on computing row canonical forms (reduced row-echelon forms) of large matrices over finite fields. In the remainder of this Introduction we summarize the contents of the paper.

In Section 2 we recall the basic theory of polynomial identities for nonassociative algebras, and describe our computational methods.

In Sections 3 to 6 we study the operations of sequential insertion and deletion, each of which has two versions depending on whether we regard the product as a set or a multiset. More precisely, we have these four operations on the monoid $\{x^p \,|\, p \geq 0\}$, which extend bilinearly to the polynomial algebra $\mathbb{F}[x]$:

$$\text{SIS:} \quad x^p x^q = x^{p+q} \tag{1}$$

$$\text{SIM:} \quad x^p x^q = (q+1) x^{p+q} \tag{2}$$

$$\text{SDS:} \quad x^p x^q = \begin{cases} x^{q-p} & \text{if } q \geq p \\ 0 & \text{if } q < p \end{cases} \tag{3}$$

$$\text{SDM:} \quad x^p x^q = \begin{cases} (q-p+1) x^{q-p} & \text{if } q \geq p \\ 0 & \text{if } q < p \end{cases} \tag{4}$$

In the 3-letter codes for the operations, S indicates sequential, I (D) indicates insertion (deletion), and S (M) indicates set (multiset).

In Sections 7 to 10 we study the operations of contex-

ISSAC'07, July 29–August 1, 2007, Waterloo, Ontario, Canada.

tual insertion and deletion, which depend on the contextual parameter w (a nonnegative integer). The string x^q must contain x^w as a substring in order for the operation to be successful; the sequential operations are the $w = 0$ case of the contextual operations. The contextual operation reduces to the sequential operation if $q \geq w$ but returns 0 otherwise (the zero polynomial in $\mathbb{F}[\Sigma]$ corresponds to the empty subset of Σ^*):

$$\text{CIS:} \qquad x^p x^q = \begin{cases} x^{p+q} & \text{if } q \geq w \\ 0 & \text{if } q < w \end{cases} \tag{5}$$

$$\text{CIM:} \qquad x^p x^q = \begin{cases} (q-w+1)x^{p+q} & \text{if } q \geq w \\ 0 & \text{if } q < w \end{cases} \tag{6}$$

$$\text{CDS:} \qquad x^p x^q = \begin{cases} x^{q-p} & \text{if } q \geq p+w \\ 0 & \text{if } q < p+w \end{cases} \tag{7}$$

$$\text{CDM:} \qquad x^p x^q = \begin{cases} (q-p-w+1)x^{q-p} & \text{if } q \geq p+w \\ 0 & \text{if } q < p+w \end{cases} \tag{8}$$

In the 3-letter codes, C indicates contextual.

Section 11 gives some directions for further research.

2. COMPUTATIONAL METHODS

The standard reference on nonassociative algebra is the book by Zhevlakov, Slinko, Shestakov and Shirshov [10]; the first chapter has a detailed discussion of polynomial identities. A recent survey article on nonassociative algebra is Bremner, Murakami and Shestakov [3]; the last section discusses computational methods.

In general, a nonassociative monomial consists of a fully parenthesized string of variables, and a nonassociative polynomial is a linear combination of nonassociative monomials. We say that a nonassociative polynomial I in n variables is a **polynomial identity** for the algebra A if $I(x_1, \ldots, x_n) = 0$ for all $x_1, \ldots, x_n \in A$. In this paper, we restrict attention to polynomials which are **homogeneous** (every monomial has the same degree) and **multilinear** (in every monomial each variable occurs exactly once). This restriction is motivated by the following fact.

Lemma 1. *Over any field of characteristic 0, every polynomial identity (not necessarily homogeneous or multilinear) is equivalent to a finite set of homogeneous multilinear polynomial identities.*

Proof. See Chapter 1 of [10]. ☐

We therefore define a **nonassociative monomial of degree** n to be a permutation of the n variables $a_1, \ldots, a_n$ together with $n-1$ pairs of parentheses indicating the association type: the order in which the binary products are to be evaluated. We write B_n for the set of all nonassociative monomials of degree n. The **space of nonassociative polynomials** in degree n is the vector space P_n with basis B_n over the field $\mathbb{F}$. There are $n!$ distinct permutations of the variables, and the Catalan number

$$C_n = \frac{1}{n}\binom{2n-2}{n-1},$$

gives the number of ways to place parentheses in a product of n factors. The dimension of P_n is therefore

$$M_n = \dim P_n = |B_n| = n! C_n = \frac{(2n-2)!}{(n-1)!}.$$

Example 1: The 12 nonassociative monomials in degree 3:

$$(ab)c,\ (ac)b,\ (ba)c,\ (bc)a,\ (ca)b,\ (cb)a,$$
$$a(bc),\ a(cb),\ b(ac),\ b(ca),\ c(ab),\ c(ba).$$

The symmetric group S_n acts naturally on B_n by permuting the variables: omitting the parentheses determining the association type, we have

$$\sigma(a_{i_1} \cdots a_{i_n}) = a_{\sigma(i_1)} \cdots a_{\sigma(i_n)}, \text{ for } \sigma \in S_n.$$

This action of S_n does not change the association type. Extending this action linearly gives P_n the structure of a representation of S_n.

Lemma 2. *The subspace of P_n consisting of the polynomial identities satisfied by an algebra A is a representation of S_n.*

Proof. Since applying $\sigma \in S_n$ to a polynomial $I \in P_n$ simply permutes the variables, it is clear that if I is an identity for A then so is σI. It is also clear that any linear combination of identities for A is again an identity for A. ☐

Let Q be a subspace of P_n which is a representation of S_n in its own right. We say that the polynomials $I_1, \ldots, I_k \in Q$ form a **set of generators** for Q if every polynomial in Q is a linear combination of the $n!k$ polynomials σI_ℓ for $\sigma \in S_n$ and $1 \leq \ell \leq k$.

2.1 Algorithm 1: Lifting identities

Suppose the algebra A satisfies the known identity $I \in P_n$, and we want to find the identities in P_{n+1} that follow from I. Introducing a new variable a_{n+1}, we lift $I = I(a_1, \ldots, a_n)$ to degree $n+1$ in $n+2$ different ways:

1. Left-multiply by a_{n+1}: $a_{n+1} I(a_1, \ldots, a_n)$.
2. Substitute $a_\ell a_{n+1}$ for a_ℓ: $I(a_1, \ldots, a_\ell a_{n+1}, \ldots, a_n)$.
3. Right-multiply by a_{n+1}: $I(a_1, \ldots, a_n) a_{n+1}$.

Example 2: The commutative identity $ab - ba$ in degree 2 can be lifted to degree 3 in 4 different ways:

left multiplication:	$c(ab) - c(ba)$	
substitution:	$(ac)b - b(ac)$,	$a(bc) - (bc)a$
right multiplication:	$(ab)c - (ba)c$	

More generally, given k identities $I_1, \ldots, I_k \in P_n$, we obtain $k(n+2)$ identities in P_{n+1}. We need to find a basis for the representation of S_{n+1} generated by these identities.

2.2 Algorithm 2: Finding a basis

Suppose we have polynomials $I_1, \ldots, I_k \in P_n$ generating a representation $Q \subseteq P_n$. To find a basis for Q, we do the following:

```
1. Create a matrix $X$ with $M_n + n!$ rows and $M_n$ columns, initialized to 0. The columns of $X$ correspond bijectively to the basis monomials $B_n$.
2. For each $\ell = 1, \ldots, k$ do:
   (a) Set $i = M_n$.
   (b) For each $\sigma \in S_n$ do:
       i. Increment $i$.
       ii. Apply $\sigma$ to $I_\ell$, obtaining $\sigma I_\ell$.
```

        iii. Store the coefficients of σI_ℓ in row i of X.

    (c) Compute the row canonical form of X. The rank cannot exceed M_n, and so the last $n!$ rows are 0.

After termination of Algorithm 2, the nonzero rows of the matrix X form a basis of the representation $Q \subseteq P_n$ generated by $I_1, \ldots, I_k$.

Example 3: Continuing from Examples 1 and 2, we apply all 6 permutations of a, b, c to the left multiplication identity and store the results in a 6×12 matrix:

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0
\end{bmatrix}$$

The row canonical form has three nonzero rows:

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}$$

We apply all permutations of a, b, c to the first substitution identity:

$$\begin{bmatrix}
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}$$

We stack together the last two matrices and compute the row canonical form:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix} \tag{9}$$

We continue with the second substitution identity and the right multiplication identity, but the rank does not increase: the representation generated by the four liftings of the commutative identity is generated by the first two liftings. This representation has dimension 9, and a basis is given by the rows of matrix (9).

2.3 Algorithm 3: Finding all identities

Suppose we want to find all the polynomial identities of degree n satisfied by a nonassociative operation on $\mathbb{F}[x]$. To find candidate identities we do the following:

1. Choose a sufficiently large but conveniently small positive integer d. Our computations will only consider the first d powers of x in the algebra $\mathbb{F}[x]$.

2. Create a matrix E (the evaluation matrix) with M_n+d rows and M_n columns, initialized to zero. The columns of E are labeled by the basis monomials B_n. The d rows of E with indices from M_n+1 to M_n+d are labeled by the basis elements $1, x, x^2, \ldots, x^{d-1}$ of $\mathbb{F}[x]$.

3. Choose sufficiently large but conveniently small positive integers s (to control the next loop) and r (to control the random number generator).

4. Repeat the following steps until the rank of E has not increased for s consecutive iterations:
    (a) Generate n random vectors of length d with integral components between 0 and $r-1$. Assign these vectors to the monomial variables $a_1, \ldots, a_n$.
    (b) Evaluate each of the M_n nonassociative monomials on the n random vectors. If this evaluation produces nonzero coefficients for powers x^e with $e \geq d$, we ignore these terms so that the result is another vector of length d.
    (c) We now have M_n linear combinations of the basis elements 1, x, x^2, $\ldots$, x^{d-1}. For each j from 1 to M_n, put the $d \times 1$ column vector of coefficients obtained from the evaluation of monomial j into rows M_n+i $(1 \leq i \leq d)$ of column j.
    (d) Compute the row canonical form of the matrix E. The rank cannot exceed M_n, and so the last d rows are zero.

5. Use the row canonical form to compute a basis for the nullspace of E.

6. Sort the basis vectors for the nullspace so that x precedes y if and only if x has fewer nonzero components than y (the identity corresponding to x has fewer terms than the identity corresponding to y).

We give a formal justification of this algorithm. Denote the basis monomials in the set B_n by T_j for $1 \leq j \leq M_n$. Consider the general polynomial identity in degree n:

$$I = \sum_{j=1}^{M_n} c_j T_j, \quad c_j \in \mathbb{F}.$$

Consider n random vectors

$$V_k = [v_{k1}, \ldots, v_{kd}] \in \mathbb{F}^d \text{ for } 1 \leq k \leq n;$$

we identify V_k with the polynomial

$$\sum_{i=1}^{d} v_{ki} x^{i-1}.$$

We evaluate the monomial T_j by setting $a_k = V_k$ for $1 \leq k \leq n$ and using the nonassociative operation on $\mathbb{F}[x]$ to evaluate each binary product. The result is a polynomial represented by the coefficient vector

$$W_j = [w_{j1}, \ldots, w_{jd}].$$

We put W_j as a column vector into rows M_n+1 to M_n+d of column j of E. After all the monomials have been evaluated and stored, each of the d rows of E with indices M_n+i $(1 \leq i \leq d)$ expresses the condition that when the linear combination I of the M_n monomials is evaluated, the coefficient of x^{i-1} must be zero. The nonzero rows of the row canonical form express linear constraints on the coefficients of the polynomial identity I. When we have done enough iterations of step 4 of the algorithm, we can be confident that we have generated all possible linear constraints, and so the nullspace will contain identities satisfied by the algebra.

(We need to check each of these identities independently, either by verifying that it is indeed satisfied by all possible substitutions of generic algebra elements, or by substituting further random algebra elements.)

Example 4: Operation (1) says that $x^p x^q = x^{p+q}$, which gives the familiar (commutative associative) product on $\mathbb{F}[x]$. For any polynomials $a, b, c \in \mathbb{F}[x]$ every nonassociative monomial in degree 3 produces the same result. Therefore the row canonical form of the matrix E will be

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Further iterations of step 4 of the algorithm will not increase the rank. From this we see that the rows of the following matrix form a basis for the nullspace.

$$\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \qquad (10)$$

We need to determine whether any of the identities represented by these rows are new (that is, do not follow from commutativity). Example 3 shows that the dimension of the space of lifted identities is 9, but matrix (10) has 11 nonzero rows, and so there are new identities: the quotient space of new identities in degree 3 has dimension 2.

2.4 Algorithm 4: Finding generators

This algorithm is similar to Algorithm 2, with the important difference that instead of starting with the zero matrix, we start with the matrix containing a basis for the lifted identities from the previous degree. Let R denote the representation of all identities in degree n which are consequences of identities of degree $n-1$. Suppose we have identities $I_1, \dots, I_k$ in degree n, which form a basis of the representation Q of all identities in degree n satisfied by the current operation. We then have the containments $R \subseteq Q \subseteq P_n$. To find generators for the quotient representation Q/R, we do the following:

1. Create a matrix X with $M_n + n!$ rows and M_n columns.
2. Let ℓ be the dimension of the representation R (so $\ell \le M_n$), and let $J_1, \dots, J_\ell$ be the identities corresponding to the (nonzero) rows of the matrix computed by Algorithm 2. Initialize rows $1, \dots, \ell$ of X to $J_1, \dots, J_\ell$.
3. Set $r = \ell$, the rank of X.
4. For each $j = 1, \dots, k$ do:
   (a) Set $i = M_n$.
   (b) For each $\sigma \in S_n$ do:
      i. Increment i.
      ii. Apply σ to I_j, obtaining σI_j.
      iii. Store the coefficients of σI_j in row i of X.
   (c) Compute the row canonical form of X. Let s be the new rank of X.
   (d) If $r < s$ then identity I_j has increased the rank, so it is not a consequence of the lifted identities combined with the previously processed identities $I_1, \dots, I_{j-1}$. Record I_j as a new generator.
   (e) Set $r = s$.

Example 5: We perform Algorithm 4 on the space of lifted identities given by matrix (9) from Example 3 and the space of all identities given by matrix (10) from Example 4. Applying all permutations of a, b, c to row 1 of (10) gives this matrix:

$$\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \qquad (11)$$

We stack matrix (9) on top of matrix (11) and compute the row canonical form. The result is matrix (9) again: the rank has not increased; row 1 of matrix (10) is not a new identity. We get the same result when we process row 2 of matrix (10). Applying all permutations of a, b, c to row 3 of matrix (10) gives:

$$\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix} \qquad (12)$$

We stack matrix (9) on top of matrix (12) and compute the row canonical form:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix} \qquad (13)$$

The rank has increased: row 3 of matrix (10) is a new identity. Matrices (10) and (13) are row equivalent since they have the same row space. Hence the result will not change as we process rows 4 to 11 of matrix (10). The identities in degree 3 are consequences of the commutative identity in degree 2 together with the new identity represented by row 3 of matrix (10). This new identity is $-(ab)c + b(ca)$, which is equivalent modulo the liftings of the commutative identity (and a permutation of the variables) to the associative identity $(ab)c - a(bc)$.

2.5 Algorithm 5: Checking equivalence

We say that two identities I_1 and I_2 in degree n are **equivalent** if each identity is in the subrepresentation of S_n generated by the lifted identities from degree $n-1$ and the other identity. To verify this equivalence we do the following:

1. Create a matrix X with $M_n+n!$ rows and M_n columns.

2. Let ℓ be the dimension of the space of lifted identities from degree $n-1$. Initialize rows $1, \ldots, \ell$ of X to a basis of the space of lifted identities.

3. Set $r = \ell$, the rank of X.

4. For each $j = 1, 2$ do:

    (a) Set $i = M_n$.

    (b) For each $\sigma \in S_n$ do: Increment i. Apply σ to I_j, obtaining σI_j. Store the coefficients of σI_j in row i of X.

    (c) Compute the row canonical form of X. Let s be the new rank of X. Set $r = s$.

5. If the rank increased between $j = 1$ and $j = 2$, then I_2 is not a consequence of I_1. If the rank did not increase, then I_2 is a consequence of I_1.

6. We now reverse the order of I_1 and I_2 and repeat steps 1 to 5 to determine whether or not I_1 is a consequence of I_2.

7. If each identity is a consequence of the other, then they are equivalent.

2.6 Rational and modular arithmetic

In principle, we prefer to do computations over the field $\mathbb{Q}$ of rational numbers. However, during the computation of the row canonical form of a large matrix (even one with small integer entries), we can obtain matrix entries with extremely large numerators and denominators. Processing these large integers requires arbitrarily large amounts of memory and slows down the computation. We avoid this difficulty by using modular arithmetic: we choose a prime number p and work over the field $\mathbb{F}_p$ of congruence classes modulo p. Each matrix entry is then represented by an integer between 0 and $p-1$, and so we have an absolute bound on the amount of memory that we need. We take p bigger than the degree of the polynomial identities: by Chapter 1 of [10], we know that linearization of identities of degree n requires division by $n!$. If we take $p < 2^8 = 256$, then each matrix entry will fit in one byte. At the end of the computation, we have polynomial identities with modular coefficients. We have to determine which rational numbers correspond to the modular coefficients; this is the price we pay for saving memory. If we are lucky, the coefficients will be congruent to small integers, or to rational numbers with small numerators and denominators: for example, with $p = 101$ we have

$$100 \equiv -1, \quad 99 \equiv -2, \quad 98 \equiv -3, \quad 51 \equiv \frac{1}{2}, \quad 50 \equiv -\frac{1}{2}.$$

In this paper all the modular coefficients were in the set $\{0, 1, 2, 3, 98, 99, 100\}$ which made their reinterpretation as rational numbers very easy. Once we have a hypothetical identity with rational coefficients for an operation, we can write another program to check it using arithmetic in characteristic 0. The computations described in the rest of this paper were done on an IBM ThinkPad T43 using Maple 8, especially `LinearAlgebra[Modular]` with $p = 101$.

3. SEQUENTIAL INSERTION: SET

No matter where we insert the string x^p into the string x^q, we obtain x^{p+q}. If we regard the result as a set (not a multiset), we obtain operation (1). This is the familiar (commutative associative) multiplication on $\mathbb{F}[x]$.

THEOREM 3. *Every polynomial identity satisfied by operation (1) on the vector space $\mathbb{F}[x]$ is a consequence of the commutative and associative identities*

$$ab - ba = 0, \qquad (ab)c - a(bc) = 0.$$

PROOF. Examples 1–5 in Section 2 show how to verify this computationally for degrees 2 and 3. We extend the computations to degrees 4 and 5, but we do not obtain any new identities. Here is a direct proof for all degrees: Any polynomial identity for a commutative associative algebra can be regarded as an element of a free commutative associative algebra (a polynomial algebra). Recalling Lemma 1, we see that the only homogeneous multilinear polynomial of degree n is the product $a_1 \cdots a_n$. It is clear that operation (1) does not satisfy the identity $a_1 \cdots a_n = 0$ for any n. ☐

In the remainder of this paper we perform all computations over the finite field $\mathbb{F}_{101}$. Since we only consider homogeneous multilinear identities, the spaces of identities in degree n have the structure of a representation of the symmetric group S_n. By the representation theory of S_n, we know that the group algebra $\mathbb{F}S_n$ decomposes in the same way (as a direct sum of full matrix algebras) for any field $\mathbb{F}$ with characteristic 0 or $p > n$. We make this assumption on $\mathbb{F}$ for the rest of the paper. It follows that the identities for the operations will be the same over $\mathbb{F}_{101}$ and over $\mathbb{Q}$ (and over any field of characteristic $p > 5$ since we only consider identities of degree ≤ 5).

4. SEQUENTIAL INSERTION: MULTISET

There are $q+1$ different ways to insert x^p into x^q: before the first occurrence of x, or after the i-th occurrence of x for $i = 1, \ldots, q$. The result is always the string x^{p+q}; if we regard this as a multiset (not a set), we obtain operation (2). Extending this bilinearly to $\mathbb{F}[x]$ gives a new nonassociative structure on the polynomial algebra.

THEOREM 4. *Every identity of degree ≤ 5 satisfied by operation (2) on the vector space $\mathbb{F}[x]$ is a consequence of the right commutative identity*

$$(ab)c - (ac)b = 0,$$

and the left symmetric identity

$$(ab)c - (ba)c - a(bc) + b(ac) = 0.$$

PROOF. For degree 2, it is clear that the operation is neither commutative ($ab - ba = 0$) nor anticommutative ($ab + ba = 0$) nor trivial ($ab = 0$); any identity in degree 2 must be equivalent to one of these. We next perform Algorithm 3 with $n = 3$. Any identity of degree 3 must be a linear combination of the 12 monomials in Example 1. The first three random vectors modulo 101 are

$$x = [70, 76, 37, 82, 29, 56], \quad y = [42, 47, 21, 41, 85, 35],$$
$$z = [15, 97, 60, 39, 11, 14].$$

After evaluating the 12 monomials, we obtain the following 6×12 matrix:

$$\begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 42 & 42 & 87 & 87 & 92 & 92 & 55 & 65 & 100 & 99 & 14 & 3 \\ 32 & 32 & 3 & 3 & 42 & 42 & 10 & 62 & 82 & 95 & 72 & 33 \\ 24 & 24 & 26 & 26 & 16 & 16 & 92 & 48 & 94 & 2 & 40 & 93 \\ 58 & 58 & 1 & 1 & 14 & 14 & 16 & 65 & 60 & 96 & 21 & 8 \\ 72 & 72 & 67 & 67 & 47 & 47 & 40 & 11 & 35 & 25 & 87 & 5 \end{bmatrix}$$

We compute the row canonical form (recall that $100 \equiv -1$ modulo 101):

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 & 100 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

We generate another three random vectors, evaluate the monomials, and obtain another 6×12 matrix. We stack together the last matrix and this new matrix. The row canonical form is the same: the matrix reached full rank after the first iteration, but we perform another 10 iterations to be sure. Extracting a basis for the nullspace from the row canonical form gives the rows of this matrix:

$$\begin{bmatrix} 100 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 100 & 0 & 0 & 0 & 100 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 100 & 0 & 0 & 100 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 100 & 0 & 0 & 0 & 0 & 100 & 0 & 1 \end{bmatrix}$$

These vectors correspond to the following six identities:

$$\begin{aligned} &-(ab)c + (ac)b = 0, && (ab)c - (ba)c - a(bc) + b(ac) = 0, \\ &-(ba)c + (bc)a = 0, && (ab)c - (ca)b - a(cb) + c(ab) = 0, \\ &-(ca)b + (cb)a = 0, && (ba)c - (ca)b - b(ca) + c(ba) = 0. \end{aligned}$$

The three identities in the first column are permuted forms of the right commutative identity. In the second column, the first is the left symmetric identity, and the second (respectively third) becomes the left symmetric identity after applying the right commutative identity to the first term (respectively the first two terms). It follows that operation (2) satisfies the right commutative and left symmetric identities, and that every identity of degree 3 is a consequence of these two identities. We now extend these computations to degrees 4 and 5 using Algorithms 1–4, but every identity in these degrees is a consequence of the repeated liftings of the right commutative and left symmetric identities. □

The right commutative and left symmetric identities define the variety of **Novikov algebras**; for further information, see Dzhumadildaev and Löfwall [5] and Osborn and Zelmanov [8].

5. SEQUENTIAL DELETION: SET

In order to delete x^p from x^q we must have $p \le q$, and we obtain x^{q-p}. If we regard the result as a set (not a multiset), we obtain operation (3): the empty string and the empty set correspond respectively to the constant polynomial 1 and the zero polynomial 0.

Theorem 5. *There are no identities in degree ≤ 2 for operation (3). The space of identities in degree 3 has dimension 3; as a representation of S_3 it is generated by the left commutative identity*

$$a(bc) - b(ac) = 0.$$

The liftings of this identity to degree 4 generate a representation of S_4 of dimension 56. The space of identities in degree 4 has dimension 60; the quotient space of new identities has dimension 4 and is generated as a representation of S_4 by this 12-term identity:

$$\begin{aligned} &((ab)c)d - ((ac)b)d - ((ba)c)d + ((bc)a)d + ((ca)b)d \\ &- ((cb)a)d - (ab)(cd) + (ac)(bd) + (ba)(cd) - (bc)(ad) \\ &- (ca)(bd) + (cb)(ad) = 0. \end{aligned}$$

The liftings of the previous two identities to degree 5 generate a representation of S_5 of dimension 1155. The space of identities in degree 5 has dimension 1180; the quotient space of new identities has dimension 25 and is generated as a representation of S_5 by these two 6-term identities:

$$\begin{aligned} &((ab)c)(de) - ((ab)d)(ce) - ((cb)a)(de) + ((cb)d)(ae) \\ &+ ((db)a)(ce) - ((db)c)(ae) = 0, \end{aligned}$$

$$\begin{aligned} &(((ab)c)d)e - (((ad)c)b)e - (((cb)a)d)e + (((cd)a)b)e \\ &- (ab)((cd)e) + (cb)((ad)e) = 0. \end{aligned}$$

Proof. For degree ≤ 3, the argument is very similar to the proof of Theorem 4. For degree 4, we start by using Algorithm 1 to find the 5 liftings of the left commutative identity to degree 4. We then use Algorithm 2 to find a basis for the representation of S_4 generated by these liftings; for this we use a matrix of size 144×120 since $M_n = 120$ and $n! = 24$. We next use Algorithm 3 with another matrix of the same size to find all the identities satisfied by operation (3) in degree 4. We take $d = 24$, so we are considering only polynomials of degree < 24. We set the random number generator to produce uniformly distributed integers $0 \le r \le 100$ (elements of the field $\mathbb{F}_{101}$). After the final rank of 60 is obtained, we perform another 20 iterations to make sure that the rank has stabilized. We finally use Algorithm 4 to compare all the identities against the lifted identities and to extract the new 12-term identity. For degree 5, there are 42 lifted identities obtained from the new 12-term identity and the 5 liftings of the left commutative identity. We repeat the same steps as in degree 4; in degree 5 we use a matrix of size 1800×1680 since $M_n = 1680$ and $n! = 120$. (The computations in fact gave $a(((db)c)e)$ for the last term in the first identity of degree 5, but we know by the left commutative identity that this equals $((db)c)(ae)$, and this gives a more symmetric identity.) We now use a separate Maple program using rational arithmetic to verify that these identities are satisfied by operation (3) when the arguments are random vectors. □

The computational proofs of the results in the following sections are similar to that of Theorem 5; for this reason we omit proofs in the rest of the paper.

6. SEQUENTIAL DELETION: MULTISET

There are $q - p + 1$ ways of deleting x^p from x^q, since x^p can begin at any position $i = 1, \ldots, q - (p - 1)$ of x^q. If we regard the result as a multiset (not a set), we obtain operation (4).

Theorem 6. *There are no identities of degree ≤ 3 for operation (4). The space of identities in degree 4 has dimension 4; it is generated as a representation of S_4 by this 6-term identity:*

$$\begin{aligned} &a(b(cd)) - a(c(bd)) - b(a(cd)) + b(c(ad)) + c(a(bd)) \\ &- c(b(ad)) = 0. \end{aligned}$$

The liftings of this identity to degree 5 generate a representation of S_5 of dimension 115. The space of identities in degree 5 has dimension 125; the quotient space of new identities has dimension 10 and is generated as a representation of S_5 by this 14-term identity:

$$\begin{aligned} &2a(b(c(de))) - a(b(d(ce))) - 2a(c(b(de))) + a(c(d(be))) \\ &- b(a(c(de))) + b(a(d(ce))) + b(c(d(ae))) - b(d(a(ce))) \\ &+ c(a(b(de))) - c(a(d(be))) - c(b(d(ae))) + c(d(a(be))) \\ &+ d(a(b(ce))) - d(a(c(be))) = 0. \end{aligned}$$

7. CONTEXTUAL INSERTION: SET

Suppose we fix integers $u, v \ge 0$ and insert the string x^p into the string x^q, subject to the condition that the insertion point must be preceded by the substring x^u and followed by the substring x^v. This operation is **contextual insertion** with parameters (u, v). There are two possible results, depending on whether x^u and x^v occur in x^q or not, and the result depends only on $w = u + v$. If we regard the result as a set (not a multiset), we obtain operation (5). We consider only the cases $1 \le w \le 4$; over this range the results do not depend on w.

Theorem 7. *For $1 \le w \le 4$, operation (5) satisfies no identities of degree ≤ 2. The space of identities in degree 3 has dimension 6; it is generated as a representation of S_3 by the left and right commutative identities:*

$$a(bc) - b(ac) = 0, \qquad (ab)c - (ac)b = 0.$$

The liftings of these identities to degree 4 generate a representation of S_4 of dimension 106. The space of identities in degree 4 also has dimension 106; there are no new identities in degree 4. The liftings of these identities to degree 5 generate a representation of S_5 of dimension 1650. The space of identities in degree 5 also has dimension 1650; there are no new identities in degree 5.

It is natural to conjecture that the same result holds for arbitrary $w \ge 1$.

8. CONTEXTUAL INSERTION: MULTISET

If we regard the result of contextual insertion as a multiset (not a set), we obtain operation (6). For this operation, the results depend on w.

8.1 Context $w = 1$

Theorem 8. *For $w = 1$, operation (6) satisfies no identities of degree ≤ 2. The space of identities in degree 3 has dimension 6; it is generated as a representation of S_3 by the right commutative and left symmetric identities of Theorem 4. The liftings of these identities to degree 4 generate a representation of S_4 of dimension 100. The space of identities in degree 4 also has dimension 100; there are no new identities in degree 4. The liftings of these identities to degree 5 generate a representation of S_5 of dimension 1610. The space of identities in degree 5 also has dimension 1610; there are no new identities in degree 5.*

8.2 Context $w = 2$

Theorem 9. *For $w = 2$, operation (6) satisfies no identities of degree ≤ 2. The space of identities in degree 3 has dimension 3; it is generated as a representation of S_3 by the right commutative identity of Theorem 4. The liftings of this identity to degree 4 generate a representation of S_4 of dimension 56. The space of identities in degree 4 has dimension 72; the quotient space of new identities has dimension 16 and is generated as a representation of S_4 by the 6-term identity of Theorem 6 together these two identities:*

$$(a(bc))d - (a(bd))c - (b(ac))d + (b(ad))c = 0,$$

$$(a(bc))d - (b(ac))d - a((bc)d) + b((ac)d) = 0.$$

The liftings of the four previous identities to degree 5 generate a representation of S_5 of dimension 1400. The space of identities in degree 5 has dimension 1445; the quotient space of new identities has dimension 45 and is generated as a representation of S_5 by the following three identities with (respectively) 10, 18 and 26 terms:

$$\begin{aligned} &((a(bc))d)e - ((a(dc))b)e - (a((bc)d))e + (a((dc)b))e \\ &+ (a(b(dc)))e + (a(b(de)))c - (a(d(bc)))e - (a(d(be)))c \\ &- a(b((dc)e)) + a(d((be)c)) = 0, \end{aligned}$$

$$\begin{aligned} &((a(bc))d)e - ((a(be))d)c - ((a(cb))d)e + ((a(ce))d)b \\ &+ ((a(eb))d)c - ((a(ec))d)b - (a(b(ce)))d + (a(b(ec)))d \\ &+ (a(c(be)))d - (a(c(eb)))d - (a(e(bc)))d + (a(e(cb)))d \\ &+ b(a(c(ed))) - b(a(e(cd))) - c(a(b(ed))) + c(a(e(bd))) \\ &+ e(a(b(cd))) - e(a(c(bd))) = 0, \end{aligned}$$

$$\begin{aligned} &2((a(bc))d)e - 2((a(bd))c)e + 2((a(db))c)e - 2((a(dc))b)e \\ &- 2((b(ca))d)e + 2((b(cd))a)e + 2((c(da))b)e - 2((c(db))a)e \\ &+ 2(a(b(dc)))e - 2(a(c(db)))e - 2(a(d(bc)))e + 2(a(d(cb)))e \\ &- 2(b(a(cd)))e + 2(b(c(ad)))e + 2(c(a(db)))e - 2(c(b(da)))e \\ &- 2(c(d(ab)))e + 2(c(d(ba)))e + 3a(c(b(de))) - 3a(c(d(be))) \\ &- b(a(d(ce))) + b(c(d(ae))) - 3c(a(b(de))) + 3c(a(d(be))) \\ &- d(a(b(ce))) + d(c(b(ae))) = 0. \end{aligned}$$

8.3 Context $w \ge 3$

Theorem 10. *For operation (6) with $w = 3, 4$ the identities in degree ≤ 4 are the same as in Theorem 9. The space of identities in degree 5 has dimension 1410; the quotient space of new identities has dimension 10 and is generated as a representation of S_5 by the following 14-term identity:*

$$\begin{aligned} &2a(b(c(de))) - 2a(b(d(ce))) - a(c(b(de))) - a(c(d(be))) \\ &+ 2a(d(b(ce))) - 2b(a(c(de))) + b(a(d(ce))) + 2b(c(a(de))) \\ &- b(c(d(ae))) - c(a(b(de))) + 3c(a(d(be))) - 2c(d(a(be))) \\ &- d(a(b(ce))) + d(c(b(ae))) = 0. \end{aligned}$$

9. CONTEXTUAL DELETION: SET

Suppose we fix integers $u, v \ge 0$ and delete the string x^p from the string x^q, subject to the condition that the deleted string must be preceded by the substring x^u and followed by the substring x^v. This operation is **contextual deletion** with parameters (u, v). There are two possible results, depending on whether x^u and x^v occur in x^q or not, and the result depends only on $w = u + v$. If we regard the result as a set (not a multiset), we obtain operation (7).

9.1 Context $w = 1$

THEOREM 11. *For operation (7) with $w = 1$, the identities in degree ≤ 4 are the same as in Theorem 5: the left commutative identity and the 12-term identity generate all the identities. The liftings of these identities to degree 5 generate a representation of S_5 of dimension 1155. The space of identities in degree 5 also has dimension 1155; there are no new identities in degree 5.*

9.2 Context $w \geq 2$

THEOREM 12. *For operation (7) with $w = 2, 3, 4$, the identities in degree ≤ 3 are the same as in Theorem 5: the left commutative identity generates all the identities. The liftings of this identity to degree 4 generate a representation of S_4 of dimension 56. The space of identities in degree 4 also has dimension 56; there are no new identities in degree 4. The liftings of the left commutative identity to degree 5 generate a representation of S_5 of dimension 1055. The space of identities in degree 5 also has dimension 1055; there are no new identities in degree 5.*

10. CONTEXTUAL DELETION: MULTISET

If we regard the result of contextual deletion as a multiset (not a set), we obtain operation (8).

THEOREM 13. *For operation (8) with $1 \leq w \leq 4$ the identities are the same as in Theorem 6: the 6-term identity of degree 4 and the 14-term identity of degree 5 generate all the identities.*

These computations produced not the degree 5 identity of Theorem 6 but the degree 5 identity of Theorem 10. We used Algorithm 5 to verify that these two identities are equivalent modulo the liftings of the 6-term identity from degree 4.

11. CONCLUDING REMARKS

We also considered alternative versions of the contextual operations in which the result is x^q (not 0) when $q < w$:

$$\text{ACIS:} \quad x^p x^q = \begin{cases} x^{p+q} & \text{if } q \geq w \\ x^q & \text{if } q < w \end{cases}$$

$$\text{ACIM:} \quad x^p x^q = \begin{cases} (q - w + 1) x^{p+q} & \text{if } q \geq w \\ x^q & \text{if } q < w \end{cases}$$

$$\text{ACDS:} \quad x^p x^q = \begin{cases} x^{q-p} & \text{if } q \geq p + w \\ x^q & \text{if } q < p + w \end{cases}$$

$$\text{ACDM:} \quad x^p x^q = \begin{cases} (q - p - w + 1) x^{q-p} & \text{if } q \geq p + w \\ x^q & \text{if } q < p + w \end{cases}$$

If $w = 0$ the alternative insertion operations reduce to the corresponding sequential operations. If $w = 0$ the alternative deletion operations do not reduce to the corresponding sequential operations, so these cases need to be considered separately. Our results for these operations involved identities with very many terms (in some cases more than 100).

In this paper we have only considered identities of degree ≤ 5. A natural computational problem is to extend these calculations to higher degrees. A natural theoretical problem is to determine whether a finite set of identities implies all the identities (in all degrees) for a given operation.

We have only considered the simplest case of an alphabet with one letter: $\Sigma = \{x\}$. A very interesting problem is to consider $|\Sigma| \geq 2$; in this case it would be difficult to give a simple formula for the operations since the monoid Σ^* is not commutative.

Some of the algorithms we have described are classical. The efficiency of our computations could be improved by faster algorithms: for solving systems of linear equations, see Faugère [6]; for reconstructing rational numbers from modular results, see Collins and Encarnación [4].

Acknowledgements

We thank Ian McQuillan for helpful discussions, and the referees for useful suggestions. This research was supported by NSERC through a Discovery Grant (Bremner) and an Undergraduate Summer Research Award (Hancock).

12. REFERENCES

[1] M. R. Bremner. DNA computing, insertion of words and left-symmetric algebras. *Proceedings of Maple Conference 2005 (July 17–20, 2005, Waterloo, Ontario, Canada)*, edited by Ilias S. Kotsireas, Waterloo Maple, 2005, pages 229–242.

[2] M. R. Bremner. Jordan algebras arising from intermolecular recombination. *Communications in Computer Algebra (SIGSAM Bulletin)* 39 (2005) 106–117.

[3] M. R. Bremner, Lúcia I. Murakami and I. P. Shestakov. Nonassociative algebras. *Handbook of Linear Algebra*, edited by Leslie Hogben, Chapman & Hall / CRC, Boca Raton, 2006, pages 69-1 to 69-26.

[4] G. E. Collins and M. J. Encarnación. Efficient rational number reconstruction. *Journal of Symbolic Computation* 20 (1995) 287–297.

[5] A. Dzhumadildaev and C. Löfwall. Trees, free right-symmetric algebras, free Novikov algebras and identities. *Homology Homotopy and Applications* 4 (2002) 165–190 (electronic).

[6] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra* 139 (1999) 61–88.

[7] Lila Kari. *On Insertion and Deletion in Formal Languages.* Ph.D. Dissertation, Department of Mathematics, University of Turku, Finland, 1991.

[8] J. M. Osborn and E. Zelmanov. Nonassociative algebras related to Hamiltonian operators in the formal calculus of variations. *Journal of Pure and Applied Algebra* 101 (1995) 335–352.

[9] G. Păun, G. Rozenberg and A. Salomaa. *DNA Computing: New Computing Paradigms.* Springer-Verlag, New York, 1998.

[10] K. A. Zhevlakov, A. M. Slinko, I. P. Shestakov and A. I. Shirshov. *Rings that are Nearly Associative.* Academic Press, New York, 1982.