

Noncommutative Involutive Bases

Gareth Evans

June 27, 2004

Abstract

Gröbner Basis theory originated in the work of Buchberger [4] and is now considered to be one of the most important and useful areas of computer algebra. In 1993, Zharkov and Blinkov [13] proposed an alternative method of computing a commutative Gröbner Basis, namely the computation of an Involutive Basis.

In the mid 1980's, Mora showed [11] that Buchberger's work could be generalised for noncommutative rings. This article explores the issues surrounding the corresponding generalisation for Involutive Bases, and constructs a noncommutative involutive division which, when used with the noncommutative involutive basis algorithm, returns a noncommutative Gröbner Basis on termination.

1 Introduction

In 1965, Bruno Buchberger published his Ph.D. thesis [4] and introduced the theory of Gröbner Bases. Over the remainder of the 20th century, many strides were made in the development of the theory, helped by the rapid development of computer algebra systems over the same period, and many important applications were found in several wide-ranging branches of mathematics and engineering.

During this intense period of research, many improvements were made to the basic algorithm for computing a Gröbner Basis (which has come to be known as Buchberger's algorithm), including criteria for detecting unnecessary computations and change of basis techniques. Indeed much of the time spent researching Gröbner Bases is dedicated to ways of improving the performance of the algorithm.

In 1993, Zharkov and Blinkov [13] noticed that there were similarities between Gröbner Basis theory and work relating to Partial Differential Equations. They introduced the theory of Involutive Bases as an alternative way of obtaining a Gröbner Basis, with the property that every Involutive Basis is a (possibly redundant) Gröbner Basis with extra combinatorial structure. Tantalisingly, early indications show that computing Gröbner Bases using involutive methods may be more efficient than using the traditional methods [7].

In the mid 1980's, Mora [11] generalised the theory of Gröbner Bases to the noncommutative case. It was discovered that the theory more or less carried over without trouble, the only major change being in the consideration of S-polynomials. This article takes a first step at showing that the theory of Involutive Bases can also be generalised to the noncommutative case, giving an alternative method of computing a noncommutative Gröbner Basis.

2 Background Theory

2.1 Commutative Gröbner Bases

Consider a commutative polynomial ring $R = K[x_1, \dots, x_n]$, where K denotes a field. An ideal J is a subring of R closed under multiplication by elements of R which we usually represent by a set of generating polynomials $P = \{p_1, p_2, \dots, p_m\}$ (P is said to be a basis for J). An arbitrary polynomial $q \in R$ is a member of the ideal J if q can be expressed as a linear polynomial combination of elements of any basis P for J .

Given a basis F for an ideal J , the central idea behind Gröbner Basis theory is to use F to find a basis G for J with the property that remainders with respect to G are unique. To accomplish this goal, two algorithms are required — the first (a division algorithm) for dividing a polynomial with respect to a set of polynomials (so that we can talk about remainders with respect to a basis), and the second

(Buchberger's algorithm) for finding the basis G , a basis which we shall call a Gröbner Basis for J .

2.1.1 The Division Algorithm

Given a polynomial ring $R = K[x_1, \dots, x_n]$, fix an ordering on the indeterminates x_1, \dots, x_n (assumed from now on to be $x_1 > x_2 > \dots > x_n$), and let $<$ denote an admissible monomial ordering on the set $M = \{x_1^{d_1} \cdots x_n^{d_n} \mid d_i \in \mathbb{N}, i = 1, \dots, n\}$ of all monomials in R .

Example 2.1 Let $\alpha = x_1^{a_1} \cdots x_n^{a_n}$ and $\beta = x_1^{b_1} \cdots x_n^{b_n}$ be two monomials. In the *lexicographic ordering*, $\alpha > \beta$ iff $a_i = b_i$ for $1 \leq i \leq j < n$ and $a_{j+1} > b_{j+1}$.

Let $LT(p)$ denote the lead term of a polynomial p with respect to the given monomial ordering. For example, if $p = 5x_2x_3 + 3x_1x_3$, then $LT(p) = 3x_1x_3$ using the lexicographic ordering. Later on, we will also use $LM(p)$ and $LC(p)$ to denote the lead monomial and lead coefficient of p (respectively x_1x_3 and 3 in the above example).

Algorithm 1 The Commutative Division Algorithm

Inputs: A polynomial p and a set of polynomials $P = \{p_1, \dots, p_m\}$ over a commutative ring $R = K[x_1, \dots, x_n]$; an admissible monomial ordering $<$.

Output: The remainder r of p w.r.t. P .

BEGIN

$r = 0$;

 WHILE ($p \neq 0$) DO

$t = LT(p)$; $j = 1$; found = false;

 WHILE ($j \leq m$ AND found == false) DO

 IF ($LT(p_j) \mid t$) THEN

 found = true; $p = p - (\frac{t}{LT(p_j)})p_j$;

 ELSE

$j = j+1$;

 END_IF

 END_WHILE

 IF (found == false) THEN

$r = r + t$; $p = p - t$;

 END_IF

 END_WHILE

 RETURN r ;

END

2.1.2 Buchberger's Algorithm

Given a Gröbner Basis G , we can solve the Ideal Membership Problem for any polynomial p simply by using the above division algorithm to test whether or not the remainder of p with respect to G is zero. This cannot be done with an arbitrary basis as the remainder or normal form of a polynomial with respect to an arbitrary basis is not in general unique. We obtain a Gröbner Basis from an arbitrary basis by considering so-called S-polynomials, one for each pair of polynomials in the current basis.

Definition 2.2 The *S-polynomial* of two polynomials p_1 and p_2 is given by the following formula.

$$\text{S-Pol}(p_1, p_2) = \frac{\text{lcm}(LM(p_1), LM(p_2))}{LT(p_1)}p_1 - \frac{\text{lcm}(LM(p_1), LM(p_2))}{LT(p_2)}p_2.$$

Algorithm 2 A Basic Commutative Gröbner Basis Algorithm

Inputs: A Basis $F = \{f_1, f_2, \dots, f_m\}$ for an ideal J over a commutative ring
 $R = K[x_1, \dots, x_n]$; an admissible monomial ordering $<$.

Output: A Gröbner Basis $G = \{g_1, g_2, \dots, g_p\}$ for J .

BEGIN

Let $G = F$ and let $L = \emptyset$;

For each pair of polynomials (g_i, g_j) in G ($i < j$),

add the S-polynomial $\text{S-Pol}(g_i, g_j)$ to L ;

WHILE (L is not empty) DO

Remove the first entry s_1 from L ;

Reduce s_1 with respect to G (with the division algorithm);

If s_1 reduces to zero then do nothing;

otherwise if s_1 reduces to $r_1 \neq 0$ add r_1 to G and add all
the S-polynomials $\text{S-Pol}(g_i, r_1)$ to L ($g_i \in G$, $g_i \neq r_1$);

END_WHILE

RETURN G ;

END

Most modern computer algebra systems possess an implementation of Buchberger's algorithm (one of the most efficient can be found in Singular [9]). Numerous improvements have been made to the algorithm over the years, including the discovery of criteria for detecting needless S-polynomial calculations [3], strategies for deciding favourable ways of processing S-polynomials [8], and various methods of obtaining a Gröbner Basis with respect to one ordering from a Gröbner Basis related to a different ordering, such as the FGLM technique [5] and the Gröbner Walk [1]. But even with all of these improvements, we cannot avoid the fact that a lot of division still occurs in the algorithm, an obstacle the involutive theory tries to navigate.

2.2 Commutative Involutive Bases

2.2.1 Involutive Division

Conventionally, a monomial m_1 is divisible by a monomial m_2 if there exists a third monomial m_3 such that $m_1 = m_2 m_3$. We use the notation $m_2 \mid m_1$ to denote that m_2 is a divisor of m_1 . An involutive division I restricts these conventional divisions by requiring all variables in m_3 to be *multiplicative* for m_2 . We use the notation $m_2 \mid_I m_1$ to denote that m_2 is an involutive divisor of m_1 .

Definition 2.3 Let M denote the set of all monomials in the polynomial ring $R = K[x_1, \dots, x_n]$. An *involutive division* I on M is defined if we can assign a set of multiplicative variables $\mathcal{M}_I(u, U)$ for any monomial u in any set of monomials $U \subset M$ such that, if $\mathcal{C}_I(u, U) = \{u \times v \mid v \in M, \text{ all variables in } v \text{ are multiplicative for } u\}$ denotes the involutive cone of the monomial $u \in U$, the following two conditions are satisfied.

- If there exist two elements $u_1, u_2 \in U$ such that $\mathcal{C}_I(u_1, U) \cap \mathcal{C}_I(u_2, U) \neq \emptyset$, then either $\mathcal{C}_I(u_1, U) \subset \mathcal{C}_I(u_2, U)$ or $\mathcal{C}_I(u_2, U) \subset \mathcal{C}_I(u_1, U)$.
- If $V \subset U$, then $\mathcal{M}_I(v, U) \subseteq \mathcal{M}_I(v, V)$ for all $v \in V$.

If the involutive division determines the multiplicative variables for a monomial $u \in U$ independent of the set U , then the division is known as a *global* division. Otherwise, the division is known as a *local* division.

Example 2.4 The Pommaret division P is a global division that assigns multiplicative variables to any monomial $u = x_1^{d_1} \cdots x_n^{d_n}$, $d_i \in \mathbb{N}$, $i = 1, \dots, n$ according to the following rule: if $1 \leq j \leq n$ is the smallest integer such that $d_j > 0$, then $\mathcal{M}_P(u) = \{x_1, \dots, x_j\}$.

Algorithm 3 The Commutative Involutive Division Algorithm

When we want to find the involutive normal form of a polynomial u w.r.t. a set of polynomials U (referenced as ‘INF’ in the following algorithms), we use our involutive division I to assign multiplicative variables to the elements of U and then apply the division algorithm (Algorithm 1) with one important change: the line

IF $(LT(p_j) \mid t)$ THEN

of Algorithm 1 is changed to

IF $(LT(p_j) \mid_I t)$ THEN

in order to reflect that we are now dealing with involutive reductions.

2.2.2 Prolongations and Autoreduction

Whereas Buchberger's algorithm constructs a Gröbner Basis by using S-polynomials, the involutive algorithm constructs a Gröbner Basis by using processes known as prolongation and autoreduction.

Definition 2.5 Given a polynomial g , a *prolongation* of g is a product gx_i , where x_i is a non-multiplicative variable of $LM(g)$ with respect to some involutive division I .

Definition 2.6 Autoreduction is the process of involutively reducing each member of a set of monomials by the rest of the set until all members are involutively irreducible.

Algorithm 4 Autoreduction

Inputs: A set of polynomials $S = \{s_1, s_2, \dots, s_\sigma\}$; an involutive division I .

Output: An Autoreduced set $T = \{t_1, t_2, \dots, t_\tau\}$.

BEGIN

$T = S$;

WHILE ($\exists t_i \in T$ s.t. $INF(t_i, T \setminus t_i) \neq t_i$) DO

 IF ($INF(t_i, T \setminus t_i) \neq 0$) THEN $T = (T \setminus t_i) \cup INF(t_i, T \setminus t_i)$;

END_WHILE

RETURN T ;

END

Algorithm 5 The Commutative Involutive Basis Algorithm

Inputs: A Basis $F = \{f_1, f_2, \dots, f_m\}$ for an ideal J over a commutative ring

$R = K[x_1, \dots, x_n]$; an admissible monomial ordering $<$;

an involutive division I .

Output (in the case of termination): an Involutive Basis $G = \{g_1, g_2, \dots, g_p\}$ for J .

BEGIN

$G = \emptyset$;

WHILE ($F \neq \emptyset$) DO

$G = \text{Autoreduce}(G \cup F)$; $F = \emptyset$;

 FOR EACH $g \in G$ DO

 FOR EACH $x_i \notin \mathcal{M}_I(LM(g), G)$ DO

$f = INF(gx_i, G)$;

 IF ($f \neq 0$) THEN $F = F \cup \{f\}$;

 END_FOR

 END_FOR

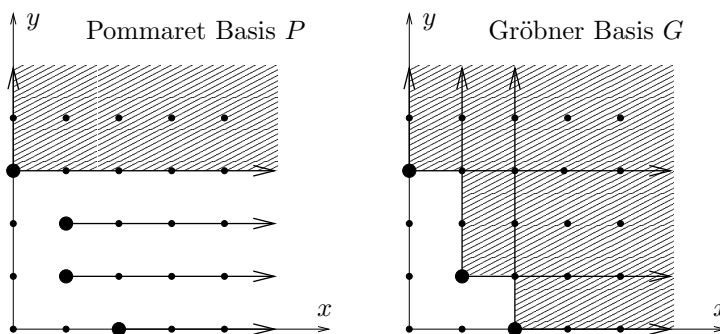
END_WHILE

RETURN G ;

END

In order for the involutive algorithm to terminate with a Gröbner Basis, the involutive division chosen must satisfy the technical criteria of continuity and constructivity [12]. In addition, if we are to guarantee the termination of the algorithm, the involutive division chosen must be Noetherian. An example of a continuous and constructive division that is not Noetherian is the Pommaret division.

Example 2.7 Consider the ideal J generated by the basis $F = \{x^2 - 2xy + 3, 2xy + y^2 + 5\}$ over the commutative ring $\mathbb{Q}[x, y]$, and let the monomial ordering be DegLex (where we order by degree first and then by Lex in the event of a tie). The Gröbner Basis is the set $G = \{x^2 + y^2 + 8, 2xy + y^2 + 5, 5y^3 - 10x + 37y\}$, and the Pommaret involutive basis is the set $P = \{5y^3 - 10x + 37y, -5xy^2 - 5x + 6y, 2xy + y^2 + 5, x^2 + y^2 + 8\}$. Noticing that the variable x is multiplicative for all polynomials in P and that the variable y is only multiplicative for the first polynomial in P , we can compare diagrams of the non-overlapping involutive cones for P and the overlapping cones for G .



An important combinatorial property of involutive bases is the fact that the involutive cones are always disjoint. One of the advantages of this is that the Hilbert function of an ideal J is easily computable with an involutive basis [2].

2.3 Noncommutative Gröbner Bases

2.3.1 Two-sided ideals

Consider a noncommutative ring $N = K[x_1, \dots, x_n]$, where K is a field. A two-sided ideal J is a subring of N closed under left *and* right multiplication by elements of N , and a noncommutative Gröbner Basis is a set of generating polynomials $G = \{g_1, \dots, g_p\}$ for a two-sided ideal J with the property that the remainder of a polynomial on division by the Gröbner Basis is unique.

To obtain a noncommutative Gröbner Basis, we need modified versions of Buchberger's algorithm and the division algorithm. The definitions of corresponding monomial orderings change as well.

Example 2.8 In the (noncommutative) lexicographical ordering, for monomials m_1 and m_2 , define $m_1 > m_2$ iff m_1 is lexicographically greater than m_2 . In other words, working left-to-right, the first

(say i -th) letter on which m_1 and m_2 differ is such that the i -th letter of m_1 is lexicographically greater than the i -th letter of m_2 in the ordering of indeterminates.

Algorithm 6 The Noncommutative Division Algorithm

When we want to find the noncommutative normal form of a polynomial p w.r.t. a set of polynomials P , we apply the division algorithm (Algorithm 1) with two changes: we change the input ring to be a noncommutative ring N , and we change the command

$$p = p - \left(\frac{t}{LT(p_j)} \right) p_j$$

of Algorithm 1 to read

$$p = p - c\ell p_j r$$

instead, where $c = \frac{LC(p)}{LC(p_j)}$ and ℓ and r are monomials chosen¹ so that $LM(p) = \ell LM(p_j)r$.

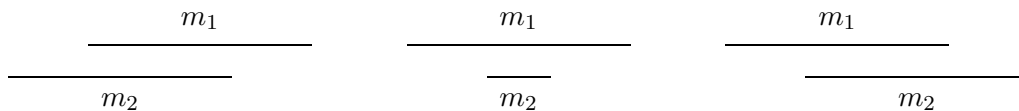
2.3.2 Overlaps

In the commutative case, there is exactly one S-polynomial for each pair of polynomials. In the noncommutative case however, there are potentially many S-polynomials for each pair of polynomials, one S-polynomial originating for each *overlap* between the lead monomials of any pair of polynomials.

Definition 2.9 Let the lead monomials of two polynomials p_1 and p_2 overlap in such a way that $\ell_1 LM(p_1)r_1 = \ell_2 LM(p_2)r_2$. If $c_1 = LC(p_2)$ and $c_2 = LC(p_1)$, then the *S-polynomial* associated with this overlap is given by the expression

$$\text{S-Pol}(p_1, p_2) = c_1 \ell_1 p_1 r_1 - c_2 \ell_2 p_2 r_2.$$

Remark 2.10 There are three possible types of overlap ('left', 'middle' and 'right'): if m_1 and m_2 are any two monomials satisfying $\deg(m_1) \geq \deg(m_2)$, then the three possible types of overlap are illustrated by the diagram shown below.



Given a pair of polynomials (p_1, p_2) , assume that $\delta = \deg(LM(p_1)) \geq \deg(LM(p_2))$. From the above, we can deduce that our pair of polynomials can have any number $0 \leq d \leq D$ of S-polynomials, where $D = 2 \times \delta - 1$, and this leads to storage problems in how we deal with S-polynomials in the noncommutative Gröbner Basis algorithm. However a far greater concern is that the algorithm is not guaranteed to terminate as it is in the commutative case, meaning that a finitely generated ideal may have an infinite noncommutative Gröbner Basis.

¹If there are several candidates for ℓ (and therefore for r), by convention choose the one with the smallest degree.

Algorithm 7 The Noncommutative Gröbner Basis Algorithm

Inputs: A Basis $F = \{f_1, f_2, \dots, f_m\}$ for an ideal J over a noncommutative ring $N = K[x_1, \dots, x_n]$; an admissible monomial ordering $<$.
Output: A Gröbner Basis $G = \{g_1, g_2, \dots, g_p\}$ for J .

BEGIN

$G = F$;

For each pair of polynomials (g_i, g_j) in G , add all the overlaps between the lead monomials $LM(g_i)$ and $LM(g_j)$ to a list L , where the k -th entry in the list is a six-tuple $T_k = (g_k, g'_k, c_k \ell_k, r_k, c'_k \ell'_k, r'_k)$ such that the following holds:

$$c_k \ell_k (LT(g_k)) r_k = c'_k \ell'_k (LT(g'_k)) r'_k;$$

WHILE (L is not empty) DO

Remove the first entry from L and compute the S-polynomial

$$s_k := \text{S-Pol}(T_k) = c_k \ell_k g_k r_k - c'_k \ell'_k g'_k r'_k;$$

Reduce s_k with respect to the current basis (using the noncommutative division algorithm). If s_k reduces to zero then do nothing;

otherwise if s_k reduces to $r_k \neq 0$ add r_k to G and add all the overlaps between r_k and elements of G to L ;

END_WHILE

END

Remark 2.11 The consideration of overlaps of leading monomials in the above algorithm is related to the overlaps of words in the Knuth-Bendix Critical Pairs Completion Algorithm from group theory — the above algorithm can be thought of as a generalisation of the Knuth-Bendix algorithm [10].

3 Noncommutative Involutive Bases

3.1 Generalising the Commutative Theory

In a noncommutative polynomial ring, a monomial m_1 is conventionally divisible by a monomial m_2 if there exist monomials m_3 and m_4 such that $m_1 = m_3 m_2 m_4$. We use the notation $m_2 \mid m_1$ to denote that m_2 is a divisor of m_1 . As in the commutative case, an involutive division will restrict these conventional divisions by using multiplicative variables, but in order to reflect the fact that left and right multiplication is not necessarily the same in a noncommutative ring, we choose to introduce the notion of left and right multiplicative variables.

Definition 3.1 For a noncommutative involutive division I , a conventional division $m_2 \mid m_1$ (where $m_1 = m_3 m_2 m_4$) is a noncommutative involutive division if all the variables in m_3 are left multiplicative

for m_2 and all the variables in m_4 are right multiplicative for m_2 . We will use the notation $m_2 \mid_I m_1$ to denote that m_2 is an involutive divisor of m_1 .

Definition 3.2 Let M denote the set of all monomials in a noncommutative polynomial ring $N = K[x_1, \dots, x_n]$, and let $U \subset M$. The *noncommutative involutive cone* of a monomial $u \in U$ is given by the expression

$$\mathcal{C}_I(u, U) = \{v_1 \times u \times v_2 \mid v_1 \in \mathcal{C}_I^L(u, U), v_2 \in \mathcal{C}_I^R(u, U)\},$$

where $\mathcal{C}_I^L(u, U) = \{v \in M \mid \text{all variables in } v \text{ are left multiplicative for } u\}$ and $\mathcal{C}_I^R(u, U) = \{v \in M \mid \text{all variables in } v \text{ are right multiplicative for } u\}$. The *involutive span* of U is given by the expression

$$\mathcal{C}_I(U) = \bigcup_{u \in U} \mathcal{C}_I(u, U).$$

Definition 3.3 Let M be as in the previous definition. A (strong) *noncommutative involutive division* I is defined on M if we can assign a set of left multiplicative variables $\mathcal{M}_I^L(u, U)$ and a set of right multiplicative variables $\mathcal{M}_I^R(u, U)$ for any monomial u in any set of monomials $U \subset M$ such that the following two conditions are satisfied.

- If there exist two elements $u_1, u_2 \in U$ such that $\mathcal{C}_I(u_1, U) \cap \mathcal{C}_I(u_2, U) \neq \emptyset$, then either $\mathcal{C}_I(u_1, U) \subset \mathcal{C}_I(u_2, U)$ or $\mathcal{C}_I(u_2, U) \subset \mathcal{C}_I(u_1, U)$.
- If $V \subset U$, then $\mathcal{M}_I^L(v, U) \subseteq \mathcal{M}_I^L(v, V)$ and $\mathcal{M}_I^R(v, U) \subseteq \mathcal{M}_I^R(v, V)$ for all $v \in V$.

If any of the above conditions are not satisfied, the involutive division is termed a *weak* division.

Definition 3.4 If the noncommutative involutive division determines the left and right multiplicative variables for a monomial $u \in U$ independent of the set U , then the division is known as a *global* division. Otherwise, the division is known as a *local* division.

3.2 Algorithms

Recall that in order to calculate a commutative involutive basis, as well as the algorithm needed to compute the basis we needed algorithms to find involutive normal forms and to perform autoreduction. Let us now consider the natural generalisations of these algorithms to the noncommutative case.

- The noncommutative normal form algorithm combines the modifications made to Algorithm 1 in Algorithms 3 and 6.
- The noncommutative autoreduction and noncommutative involutive basis algorithms are virtually identical to their commutative counterparts.

Algorithm 8 The Noncommutative Involutive Division Algorithm

Inputs: A polynomial p and a set of polynomials $P = \{p_1, \dots, p_m\}$ over a noncommutative ring $N = K[x_1, \dots, x_n]$; an admissible monomial ordering $<$; a noncommutative involutive division I .

Output: The noncommutative involutive normal form ('NCINF') r of p w.r.t. P .

```

BEGIN
  r = 0;
  WHILE (p ≠ 0) DO
    t = LT(p); j = 1; found = false;
    WHILE (j ≤ m AND found == false) DO
      IF (LT(pj) |I t) THEN
        found = true;
        p = p - clpjr, where  $c = \frac{LC(p)}{LC(p_j)}$  and  $LM(p) = \ell LM(p_j)r$ ;2
      ELSE
        j = j+1;
      END_IF
    END_WHILE
    IF (found == false) THEN
      r = r + t; p = p - t;
    END_IF
  END_WHILE
  RETURN r;
END

```

Algorithm 9 Noncommutative Autoreduction

Inputs: A set of polynomials $S = \{s_1, s_2, \dots, s_\sigma\}$; a noncommutative involutive division I .

Output: An Autoreduced set $T = \{t_1, t_2, \dots, t_\tau\}$.

```

BEGIN
  Let  $T = S$ ;
  WHILE ( $\exists t_i \in T$  s.t.  $NCINF(t_i, T \setminus t_i) \neq t_i$ ) DO
    IF ( $NCINF(t_i, T \setminus t_i) \neq 0$ ) THEN  $T = (T \setminus t_i) \cup NCINF(t_i, T \setminus t_i)$ ;
  END_WHILE
  RETURN  $T$ ;
END

```

²If there are several candidates for ℓ (and therefore for r), by convention (as before) choose the one with the smallest degree.

Algorithm 10 The Noncommutative Involutive Basis Algorithm

Inputs: A Basis $F = \{f_1, f_2, \dots, f_m\}$ for an ideal J over a noncommutative ring
 $N = K[x_1, \dots, x_n]$; an admissible monomial ordering $<$;
a noncommutative involutive division I .

Output (in the case of termination): a Noncommutative Involutive Basis G for J .

BEGIN

$G = \emptyset$;

WHILE $F \neq \emptyset$ DO

$G = \text{Noncommutative_Autoreduce}(G \cup F)$;

$F = \emptyset$;

FOR EACH $g \in G$ DO

FOR EACH $x_i \notin \mathcal{M}_I^L(LM(g), G)$ DO

$f = \text{NCINF}(x_i g, G)$;

IF $f \neq 0$ THEN $F = F \cup \{f\}$;

END_FOR

FOR EACH $x_i \notin \mathcal{M}_I^R(LM(g), G)$ DO

$f = \text{NCINF}(g x_i, G)$;

IF $f \neq 0$ THEN $F = F \cup \{f\}$;

END_FOR

END_FOR

END_WHILE

RETURN G ;

END

3.3 Open Questions

3.3.1 Gröbner Divisions

In the commutative case, in order for the involutive basis algorithm to terminate with a commutative Gröbner Basis, the involutive division used must be *continuous* and *constructive*. Let us now consider the corresponding definitions of continuity and constructivity in the noncommutative case, basing our definitions on those found in [12].

Definition 3.5 A noncommutative involutive division I is called *continuous* if for any set U of monomials and for any finite sequence $\{u_i\}_{(1 \leq i \leq k)}$ of elements in U such that for all $i < k$, either

$$\exists x_j \notin \mathcal{M}_I^L(u_i, U) \text{ such that } u_{i+1} \mid_I x_j u_i \quad (1)$$

or

$$\exists x_j \notin \mathcal{M}_I^R(u_i, U) \text{ such that } u_{i+1} \mid_I u_i x_j, \quad (2)$$

the inequality $u_i \neq u_j$ for $i \neq j$ holds.

Remark 3.6 Notions of *left continuity* and *right continuity* are obtained by requiring the sequence $\{u_i\}_{(1 \leq i \leq k)}$ to satisfy the single condition (1) or (2) respectively.

Definition 3.7 Let I be a noncommutative involutive division and U a set of monomials. Choose a monomial $u_i \in U$ and a non-multiplicative variable $x_\alpha \notin \mathcal{M}_I^L(u_i, U)$ such that:

- $x_\alpha u_i \notin \mathcal{C}_I(U)$;
- if there exists $u_j \in U$ and $x_\beta \notin \mathcal{M}_I^L(u_j, U)$ such that $x_\beta u_j \mid x_\alpha u_i$ but $x_\beta u_j \neq x_\alpha u_i$, then $x_\beta u_j \in \mathcal{C}_I(U)$.
- if there exists $u_k \in U$ and $x_\gamma \notin \mathcal{M}_I^R(u_k, U)$ such that $u_k x_\gamma \mid x_\alpha u_i$ but $u_k x_\gamma \neq x_\alpha u_i$, then $u_k x_\gamma \in \mathcal{C}_I(U)$.

The division I is *left constructive* if for any such set U and any such prolongation $x_\alpha u_i$ no monomial $u \in \mathcal{C}_I(U)$ with $x_\alpha u_i \in \mathcal{C}_I(u, U \cup u)$ exists.

Definition 3.8 Let I be a noncommutative involutive division and U a set of monomials. Choose a monomial $u_i \in U$ and a non-multiplicative variable $x_\alpha \notin \mathcal{M}_I^R(u_i, U)$ such that:

- $u_i x_\alpha \notin \mathcal{C}_I(U)$;
- if there exists $u_j \in U$ and $x_\beta \notin \mathcal{M}_I^L(u_j, U)$ such that $x_\beta u_j \mid u_i x_\alpha$ but $x_\beta u_j \neq u_i x_\alpha$, then $x_\beta u_j \in \mathcal{C}_I(U)$.
- if there exists $u_k \in U$ and $x_\gamma \notin \mathcal{M}_I^R(u_k, U)$ such that $u_k x_\gamma \mid u_i x_\alpha$ but $u_k x_\gamma \neq u_i x_\alpha$, then $u_k x_\gamma \in \mathcal{C}_I(U)$.

The division I is *right constructive* if for any such set U and any such prolongation $u_i x_\alpha$ no monomial $u \in \mathcal{C}_I(U)$ with $u_i x_\alpha \in \mathcal{C}_I(u, U \cup u)$ exists.

Definition 3.9 A noncommutative involutive division I is *constructive* if it is both left constructive and right constructive.

Open Question 1 In the case of termination, will the noncommutative involutive algorithm return a noncommutative Gröbner Basis if a continuous and constructive involutive division is used? If the answer turns out to be ‘no’, what other conditions are required in order for the answer to be affirmative?

Definition 3.10 In the absence of an answer to the above question, we shall call a noncommutative involutive division a *Gröbner division* if Algorithm 10 always returns a noncommutative Gröbner Basis in the case of termination.

3.3.2 Noncommutative Gröbner Basis \Rightarrow Noncommutative Involutive Basis?

If Algorithm 10 terminates, we now know that it returns a Gröbner Basis for the input ideal if a Gröbner division is used. Knowing whether or not the algorithm terminates is another matter altogether, and perhaps it would be a good idea at this stage to summarise the termination properties of other algorithms thus far encountered.

	Guaranteed to terminate?
Buchberger's Commutative Gröbner Basis Algorithm (Algorithm 2)	Yes
The Commutative Involutive Basis Algorithm (Algorithm 5)	Yes ³
The Noncommutative Gröbner Basis Algorithm (Algorithm 7)	No

In the case that the noncommutative Gröbner Basis algorithm fails to terminate, the noncommutative involutive basis algorithm will not terminate either as it will in effect be trying to compute the same infinite Gröbner Basis. We therefore conclude that the noncommutative involutive basis algorithm is not guaranteed to terminate.

Open Question 2 If the noncommutative Gröbner Basis algorithm terminates for an input set of polynomials generating an ideal J , will the noncommutative involutive algorithm terminate given the same input set, assuming that a Gröbner division is used?

3.4 Noncommutative Involutive Divisions

In order to show that Algorithm 10 can be used to compute noncommutative Gröbner Bases, we shall define in this section an example of a Gröbner division. We shall begin however by considering two (non-Gröbner) *extreme* global divisions corresponding to assigning no multiplicative variables and no non-multiplicative variables for any given monomial.

Definition 3.11 (The 'Empty' Division) Given any monomial m , let m have no (left or right) multiplicative variables.

Definition 3.12 (The 'Full' Division) Given any monomial m , let m have no (left or right) non-multiplicative variables (in other words, all variables are left and right multiplicative for m).

It is clear that any set of polynomials is an involutive basis using the full division — no prolongations can ever arise in the involutive algorithm — implying that the full division is not a Gröbner division. It is equally clear that the full division is a weak division as the involutive cones can never be disjoint if there is more than one element in the basis — this is easily verified by considering a common multiple of any two polynomials that are members of the basis.

³Provided a Noetherian involutive division is chosen

In contrast, the involutive algorithm will never terminate when the empty division is used as a polynomial has to be found to ‘cover’ each reducible monomial in the input ideal, of which there are an infinite number. However the empty division is a strong division as the involutive cone of any polynomial is a singleton set containing only the polynomial itself.

It is very easy to define an involutive division, but considerably harder to find a Gröbner division. In commutative involutive basis theory, there are several divisions (such as Pommaret, Thomas and Janet) that are ‘Gröbner divisions’ in the sense that they return commutative Gröbner bases in the event of termination. Unfortunately these divisions (unlike the accompanying theory) do not generalise to the noncommutative case.

Open Question 3 Are there any global noncommutative involutive divisions that are Gröbner divisions?

For our first significant noncommutative involutive division, we shall consider a division where multiplicative variables are based on overlaps of monomials. We will also need the following standard definitions.

Definition 3.13 In the noncommutative *degree-reverse-lexicographical ordering* (DegRevLex), if m_1 and m_2 are any two monomials, define $m_1 > m_2$ iff $\deg(m_1) > \deg(m_2)$ or $\deg(m_1) = \deg(m_2)$ and the last (say i -th) letter on which m_1 and m_2 differ is such that the i -th letter of m_1 is lexicographically *less* than the i -th letter of m_2 in the ordering of indeterminates.

Definition 3.14 Let $\text{Subword}(m, i, j)$ denote the subword of the monomial m starting at position i and ending at position j . For example, if $m = xyzxyz$, then $\text{Subword}(m, 2, 4) = yzx$. In a similar vein, let $\text{Prefix}(m, i)$ denote the prefix of the monomial m of length i and let $\text{Suffix}(m, j)$ denote the suffix of the monomial m of length j .

Algorithm 11 A Local Noncommutative Involutive Division

Input: A set of distinct monomials $M = \{m_1, \dots, m_p\}$ ordered by DegRevLex ($m_1 >_{\text{drl}} m_2 >_{\text{drl}} \dots >_{\text{drl}} m_p$), where $m_i \in K[x_1, \dots, x_n]$ ($x_1 > \dots > x_n$).

Output: A table T of left and right multiplicative variables for all $m_i \in M$, where each entry of T is either 1 (multiplicative) or 0 (non-multiplicative).

BEGIN

Create a table T of multiplicative variables as shown below:

	x_1^L	x_1^R	x_2^L	x_2^R	\dots	x_n^L	x_n^R
m_1	1	1	1	1	\dots	1	1
m_2	1	1	1	1	\dots	1	1
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
m_p	1	1	1	1	\dots	1	1

FOR EACH monomial $m_i \in M$ ($1 \leq i \leq p$) DO

FOR EACH monomial $m_j \in M$ ($i \leq j \leq p$) DO

Let $m_i = x_{i_1}x_{i_2}\dots x_{i_\alpha}$ and $m_j = x_{j_1}x_{j_2}\dots x_{j_\beta}$;

// Middle Overlaps

FOR EACH k ($1 \leq k \leq \alpha - \beta + 1$) DO

IF (Subword($m_i, k, k + \beta - 1$) == m_j) THEN

IF ($k + \beta < \alpha + 1$) THEN SET $T(m_j, x_{i_{k+\beta}}^R) = 0$;

ELSE SET $T(m_j, x_{i_{k-1}}^L) = 0$;

END_IF

END_FOR

// Left & Right Overlaps

FOR EACH k ($1 \leq k \leq \beta - 1$) DO

IF (Prefix(m_i, k) == Suffix(m_j, k)) THEN

IF ($T(m_i, x_{j_{\beta-k}}^L) + T(m_j, x_{i_{k+1}}^R) == 2$) THEN SET $T(m_j, x_{i_{k+1}}^R) = 0$;

END_IF

IF (Suffix(m_i, k) == Prefix(m_j, k)) THEN

IF ($T(m_i, x_{j_{k+1}}^R) + T(m_j, x_{i_{\alpha-k}}^L) == 2$) THEN SET $T(m_j, x_{i_{\alpha-k}}^L) = 0$;

END_IF

END_FOR

END_FOR

END_FOR

RETURN T ;

END

Remark 3.15 The above algorithm ensures that the common multiple associated with any overlap of two monomials is involutively irreducible by at least one of the monomials. In order to do this, certain choices are made in the algorithm regarding which monomial we choose to *block* from being an involutive divisor of the common multiple, choices which are in no way unique and do affect the table given as output and the details of the proof of the following theorem.

Theorem 3.16 *The noncommutative involutive division defined in Algorithm 11 is a Gröbner division.*

In order to prove the above theorem, we need to recall the following result from noncommutative Gröbner Basis Theory.

Definition 3.17 (Buchberger’s 2nd Criterion) Let $T_{12} = (f_1, f_2, c_1\ell_1, r_1, c_2\ell_2, r_2)$ be an arbitrary six-tuple with common multiple $\ell_1(LM(f_1))r_1 = CM = \ell_2(LM(f_2))r_2$ (f_1 and f_2 are polynomials, ℓ_1, ℓ_2, r_1 and r_2 are monomials, and c_1 and c_2 are coefficients). Suppose that the lead monomial of a third polynomial f_3 (not necessarily distinct from f_1 or f_2) divides the common multiple so that $CM = \ell_3(LM(f_3))r_3$ (note that if $f_3 = f_1$ or if $f_3 = f_2$ then $LM(f_3)$ must divide CM in a different way than the division specified in T_{12}). We can now form the six-tuples $T_{13} = (f_1, f_3, c_1\ell_1, r_1, c_3\ell_3, r_3)$ and $T_{23} = (f_2, f_3, c_2\ell_2, r_2, c_3\ell_3, r_3)$. It is apparent that there is a connection between the S-polynomials⁴ associated with T_{12}, T_{13} and T_{23} , namely

$$\begin{aligned} s_{12} &= s_{13} - s_{23}, \\ c_1\ell_1(f_1)r_1 - c_2\ell_2(f_2)r_2 &= (c_1\ell_1(f_1)r_1 - c_3\ell_3(f_3)r_3) - (c_2\ell_2(f_2)r_2 - c_3\ell_3(f_3)r_3). \end{aligned}$$

If s_{13} and s_{23} are two S-polynomials that reduce to zero, then the S-polynomial s_{12} also reduces to zero.

Proof of Theorem 3.16 We are required to show that if Algorithm 10 terminates, then it returns a noncommutative Gröbner Basis G for an input set F of generators of an ideal J over a noncommutative polynomial ring N .

To begin with, it is clear that if we can show that G is a noncommutative Gröbner Basis for some ideal J' , then it must also be a noncommutative Gröbner Basis of the ideal J . This is because the noncommutative involutive basis algorithm does not change the ideal considered due to the fact that it can only construct multiples of basis elements and can only reduce polynomials with respect to basis elements. Our goal therefore is to show that the output basis G is a noncommutative Gröbner Basis.

If Algorithm 7 (Buchberger’s Noncommutative Gröbner Basis Algorithm) is given a noncommutative Gröbner Basis G as input, then it must be the case that all S-polynomials involving elements of G reduce to zero — otherwise we reach the contradiction that G is not a Gröbner Basis. It follows that

⁴The result still holds if T_{13} and T_{23} represent expressions that are *multiples* of S-polynomials

one way of showing that a basis G is a Gröbner Basis is to show that all S-polynomials involving elements of G reduce to zero, and this is how we will now proceed.

Assume that the basis $G = \{g_1, \dots, g_p\}$ is sorted (by lead monomial) with respect to the DegRevLex monomial ordering (greatest first), and let $M = \{\text{LM}(g_1), \dots, \text{LM}(g_p)\} = \{m_1, \dots, m_p\}$ be the set of leading monomials. Let T be the table obtained by applying Algorithm 11 to G . Every zero entry $T(m_i, x_j^\Gamma)$ ($\Gamma \in \{L, R\}$) in the table corresponds to a prolongation $g_i x_j$ or $x_j g_i$ that involutively reduces to zero.

Let S be the set of S-polynomials involving elements of G , where the t -th entry of S ($1 \leq t \leq \sigma := |S|$) is the polynomial

$$s_t = c_t \ell_t g_i r_t - c'_t \ell'_t g_j r'_t,$$

where $\ell_t m_i r_t = CM = \ell'_t m_j r'_t$ is the common multiple of an overlap between the monomials $m_i, m_j \in M$. We must now show that every entry in S reduces to zero using G .

Recall that each entry in S corresponds to a particular type of overlap — ‘middle’, ‘left’ or ‘right’. If we split the middle overlaps into three further types (‘suffix’, ‘internal’ and ‘prefix’), it follows that we have five cases to deal with in total. For brevity, we will only consider the suffix case, but we will comment on how the proof generalises to the four other cases.

(1) Consider an arbitrary entry $s_t \in S$ ($1 \leq t \leq \sigma$) corresponding to an overlap where the monomial m_j is a suffix of the monomial m_i . This means that $s_t = c_t g_i - c'_t \ell'_t g_j$ for some $g_i, g_j \in G$, and so the common multiple is $m_i = CM = \ell'_t m_j$. Let $m_i = x_{i_1} \dots x_{i_\alpha}$, let $m_j = x_{j_1} \dots x_{j_\beta}$, and let $D = \alpha - \beta$.

$$\begin{array}{l} m_i = \\ m_j = \end{array} \quad \begin{array}{ccccccc} \overline{x_{i_1}} & \overline{x_{i_2}} & \text{---} & \overline{x_{i_D}} & \overline{x_{i_{D+1}}} & \overline{x_{i_{D+2}}} & \text{---} & \overline{x_{i_{\alpha-1}}} & \overline{x_{i_\alpha}} \\ & & & & \overline{x_{j_1}} & \overline{x_{j_2}} & \text{---} & \overline{x_{j_{\beta-1}}} & \overline{x_{j_\beta}} \end{array}$$

Because m_j is a suffix of m_i , then $T(m_j, x_{i_D}^L) = 0$. This gives rise to the prolongation $x_{i_D} g_j$ of g_j . But we know that all prolongations involutively reduce to zero, so there must exist a monomial $m_k = x_{k_1} \dots x_{k_\gamma} \in M$ such that m_k involutively divides $x_{i_D} m_j$. Assuming that $x_{k_\gamma} = x_{j_\kappa}$, any candidate for m_k must be a suffix of $x_{i_D} m_j$ otherwise $T(m_k, x_{j_{\kappa+1}}^R) = 0$ (because of the overlap between m_i and m_k) and so we cannot involutively multiply m_k on the right in order to form $x_{i_D} m_j$. But if m_k is a suffix of $x_{i_D} m_j$, then we must have $m_k = x_{i_D} m_j$ otherwise $T(m_k, x_{i_{\alpha-\gamma}}^L) = 0$ and so we cannot involutively multiply m_k on the left in order to form $x_{i_D} m_j$. We have therefore shown that there exists a monomial $m_k = x_{k_1} \dots x_{k_\gamma} \in M$ such that m_k is a suffix of m_i and $\gamma = \beta + 1$.

$$\begin{array}{l} m_i = \\ m_j = \\ m_k = \end{array} \quad \begin{array}{ccccccc} \overline{x_{i_1}} & \overline{x_{i_2}} & \text{---} & \overline{x_{i_D}} & \overline{x_{i_{D+1}}} & \overline{x_{i_{D+2}}} & \text{---} & \overline{x_{i_{\alpha-1}}} & \overline{x_{i_\alpha}} \\ & & & & \overline{x_{j_1}} & \overline{x_{j_2}} & \text{---} & \overline{x_{j_{\beta-1}}} & \overline{x_{j_\beta}} \\ & & & \overline{x_{k_1}} & \overline{x_{k_2}} & \overline{x_{k_3}} & \text{---} & \overline{x_{k_{\gamma-1}}} & \overline{x_{k_\gamma}} \end{array}$$

In the case $D = 1$, it is clear that $m_k = m_i$, and so the first step in the reduction of the prolongation $x_{i_1} g_j$ of g_j is to take away the multiple $(\frac{c_t}{c'_t}) g_i$ of g_i from $x_{i_1} g_j$ to leave the polynomial $x_{i_1} g_j - (\frac{c_t}{c'_t}) g_i =$

Having shown that the division defined in Algorithm 11 is a Gröbner division, let us now demonstrate that a Gröbner division need not be continuous.

Example 3.18 Let $U = \{z^2\}$. Algorithm 11 assigns multiplicative variables to U as follows:

	z^L	z^R
z^2	1	0

Let $\{u_1, u_2\} = \{z^2, z^2\}$ be a sequence of elements from U . We claim that this sequence satisfies the condition

$$\forall i < k, \exists x_j \notin \mathcal{M}_I^R(u_i, U) \text{ such that } u_{i+1} \mid_I u_i x_j,$$

where $k = 2$. To verify this claim, the only case to consider is the case $i = 1$, checking that $u_2 \mid_I u_1 x_j$ for some $x_j \notin \mathcal{M}_I^R(u_1, U)$. It is clear from the table that the only possible choice for x_j is z , and it is equally clear that $z^2 = u_2 \mid_I u_1 x_j = (z^2)z = z(z^2)$ as $z \in \mathcal{M}_I^L(z^2, U)$. We can now immediately deduce that the division defined in Algorithm 11 is not continuous as we have found a sequence satisfying the above condition in which two elements are identical ($u_1 = u_2$).

3.5 Examples

Example 3.19 Let $F = \{xy + x, -yx + x, z + x\}$ be a basis for an ideal J over the noncommutative ring $N = \mathbb{Q}[x, y, z]$ ($x > y > z$), and let our monomial ordering be DegRevLex. Let us now apply Algorithm 10 to F in order to obtain the Involutive Basis with respect to the division defined in Algorithm 11.

- `Noncommutative_Autoreduce` $\{xy + x, -yx + x, z + x\}$.
 - The polynomial $z + x$ is involutively reduced with respect to the set $\{xy + x, -yx + x\}$ using the table

	x^L	x^R	y^L	y^R	z^L	z^R
$xy + x$	1	1	1	1	1	1
$-yx + x$	0	1	1	0	1	1

to leave the polynomial $z + x$ (i.e. the polynomial is irreducible).

- Similarly, the polynomials $-yx + x$ and $xy + x$ are irreducible.
- To process the prolongations, we need to know which variables are multiplicative for our set of polynomials. This information is given by the following table.

	x^L	x^R	y^L	y^R	z^L	z^R
$xy + x$	1	1	1	1	1	1
$-yx + x$	0	1	1	0	1	1
$z + x$	1	1	1	1	1	1

- There is one left prolongation:

$$x(-yx + x) = -xyx + x^2 \rightarrow -xyx + x^2 + (xy + x)x = 2x^2 \equiv x^2.$$

- There is one right prolongation:

$$\begin{aligned} (-yx + x)y = -yxy + xy &\rightarrow -yxy + xy + y(xy + x) = xy + yx \\ &\rightarrow xy + yx - (xy + x) = yx - x \\ &\rightarrow yx - x + (-yx + x) = 0. \end{aligned}$$

- At the end of the first pass, $F = \{x^2\}$ and $G = \{xy + x, -yx + x, z + x\}$.
- Autoreducing the set $G \cup F$ does not reduce any polynomials so we now process the prolongations of the set $\{xy + x, -yx + x, x^2, z + x\}$ with the table

	x^L	x^R	y^L	y^R	z^L	z^R
$xy + x$	1	1	1	1	1	1
$-yx + x$	0	1	1	0	1	1
x^2	1	0	0	0	1	1
$z + x$	1	1	1	1	1	1

- There are 2 left prolongations:

$$\begin{aligned} x(-yx + x) = -xyx + x^2 &\rightarrow -xyx + x^2 + (xy + x)x = 2x^2 \\ &\rightarrow 2x^2 - 2(x^2) = 0; \\ y(x^2) = yx^2 &\rightarrow yx^2 + (-yx + x)x = x^2 \\ &\rightarrow x^2 - x^2 = 0. \end{aligned}$$

- There are 3 right prolongations:

$$\begin{aligned} (-yx + x)y = -yxy + xy &\rightarrow 0 \text{ (as before);} \\ (x^2)y = x^2y &\rightarrow x^2y - x(xy + x) = -x^2 \\ &\rightarrow -x^2 + x^2 = 0; \\ (x^2)x = x^3 &\rightarrow x^3 - x(x^2) = 0. \end{aligned}$$

- The algorithm now terminates with the Involutive Basis $G = \{xy + x, -yx + x, x^2, z + x\}$. Because we are using a Gröbner division, G is also a Gröbner Basis for the input ideal (by chance the minimal and reduced Gröbner Basis for J).

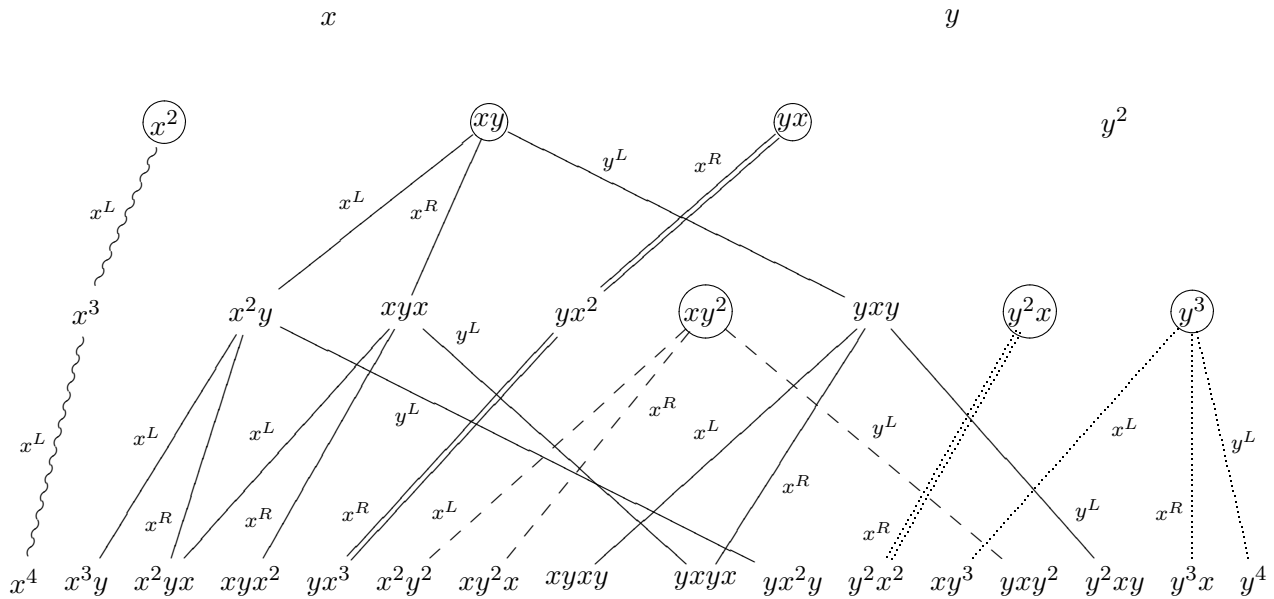
Remark 3.20 The above example shows that the division defined in Algorithm 11 is a weak division as the monomial xyz (for example) is involutively reducible by both the polynomials $xy + x$ and $z + x$ of the involutive basis, and so the involutive cones of the involutive basis are not disjoint.

Example 3.21 Let $F = \{x^2 - 2xy + 3, 2xy + y^2 + 5\}$ be a basis for an ideal J over the noncommutative ring $N = \mathbb{Q}[x, y]$ ($x > y$), and let our monomial ordering be DegLex. Applying Algorithm 10 to F in order to obtain the Involutive Basis with respect to the division defined in Algorithm 11, we obtain the involutive basis $G = \{5y^3 + 37y - 10x, 5xy^2 + 5x - 6y, 5y^2x + 5x - 6y, 2xy + y^2 + 5, 2yx + y^2 + 5, x^2 + y^2 + 8\}$ with multiplicative variables as shown below.

	x^L	x^R	y^L	y^R
$5y^3 + 37y - 10x$	1	1	1	0
$5xy^2 + 5x - 6y$	1	1	1	0
$5y^2x + 5x - 6y$	0	1	0	0
$2xy + y^2 + 5$	1	1	1	0
$2yx + y^2 + 5$	0	1	0	0
$x^2 + y^2 + 8$	1	0	0	0

In this particular example the involutive cones are disjoint, an assertion that can be verified by using finite state automata⁵ to deduce that no monomial is involutively reducible by more than one basis element. We can visualise these disjoint cones by using a diagram of the monomial lattice, part of which is shown below.

1



⁵For each lead monomial of the involutive basis, construct the automaton whose language is the set of all monomials involutively reducible by that lead monomial. One can then prove the assertion by showing that the language of any intersection automaton (constructed by using any pair of the aforementioned automata) is the empty language.

4 Conclusion

In this article we have seen how it is possible to generalise commutative involutive basis theory to the noncommutative case. We saw an algorithm that can be used to compute noncommutative involutive bases, and we constructed an involutive division that when used with the algorithm returns a noncommutative Gröbner Basis on termination.

As far as further work is concerned, apart from the questions raised in this article, there are plenty of other avenues to explore. Can the improvements made to the involutive algorithm in the commutative case (see for example [6]) be applied in the noncommutative case? Will the noncommutative involutive basis algorithm be more efficient than the noncommutative Gröbner Basis algorithm in obtaining a noncommutative Gröbner Basis? And will there be any advantages in computing noncommutative involutive bases, such as the easier computation of the Hilbert function in the commutative case?

Acknowledgements

The author is grateful to the members of the Pure Mathematics Group at the University of Wales, Bangor for valuable help and advice in preparing this article. Particular thanks go to Dr. Chris Wensley for regular discussions and insightful comments. This work has been supported by the EPSRC.

References

- [1] B. Amrhein, O. Gloor and W. Küchlin. On the Walk. *Theoret. Comput. Sci.* **187** (1–2) (1997) 179–202.
- [2] J. Apel. The Theory of Involutive Divisions and an Application to Hilbert Function Computations. *J. Symbolic Comput.* **25** (6) (1998) 683–704.
- [3] B. Buchberger. A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Bases. In *Symbolic and Algebraic Computation (EUROSAM '79, Internat. Sympos., Marseille, 1979)*, volume 72 of *Lecture Notes in Comput. Sci.*, 3–21. Springer, Berlin (1979).
- [4] B. Buchberger. An Algorithmic Criterion for the Solvability of a System of Algebraic Equations. *Translation of Ph.D. thesis by M. Abramson and R. Lumbert, in Gröbner Bases and Applications, B. Buchberger and F. Winkler, (eds) Proc. London Math. Soc.* **251**.
- [5] J. C. Faugère, P. Gianni, D. Lazard and T. Mora. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *J. Symbolic Comput.* **16** (4) (1993) 329–344.

- [6] V. Gerdt. Involutive Division Technique: Some Generalizations and Optimizations. *J. Math. Sci. (New York)* **108 (6)** (2002) 1034–1051.
- [7] V. P. Gerdt, Yu. A. Blinkov and D. A. Yanovich. Construction of Janet bases II: Polynomial Bases. In V. G. Ghanza, E. W. Mayr and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing — CASC 2001*, 249–263. Springer-Verlag, Berlin (2001).
- [8] A. Giovini, T. Mora, G. Niedi, L. Robbiano and C. Traverso. “One sugar cube, please” OR Selection strategies in Buchberger algorithm. *Proc. Int. Symp. Symbolic and Algebraic Computation (Bonn, West Germany, 1991)* 49–54.
- [9] G. M. Greuel, G. Pfister and H. Schönemann. SINGULAR 2.0.4. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern (2001). <http://www.singular.uni-kl.de>.
- [10] A. Heyworth. Rewriting as a special case of Gröbner Basis Theory. In M. Atkinson, N. D. Gilbert, J. Howie, S. A. Linton and E. F. Robertson, editors, *Computational and Geometric Aspects of Modern Algebra (Edinburgh, 1998)*, volume 275 of *Proc. London Math. Soc.*, 101–105. Cambridge Univ. Press, Cambridge (2000).
- [11] T. Mora. An Introduction to Commutative and Non-Commutative Gröbner Bases. *Theoret. Comput. Sci.* **134** (1994) 131–173.
- [12] W. M. Seiler. A Combinatorial Approach to Involution and δ -Regularity I: Involutive Bases in Polynomial Algebras of Solvable Type. *Preprint Universität Mannheim* .
- [13] A. Yu. Zharkov and Yu. A. Blinkov. Involution Approach to Investigating Polynomial Systems. *Math. Comp. Simul.* **42** (1996) 323–332.