

# Involutive Complete Rewrite Systems

Gareth Alun Evans

Mathematics Division, School of Informatics,  
University of Wales, Bangor, Gwynedd, LL57 1UT, UK  
`gevans@informatics.bangor.ac.uk`

**Abstract.** Given a monoid rewrite system  $R$ , one way of obtaining a complete rewrite system for  $R$  is to use the classical Knuth-Bendix critical pairs completion algorithm. It is well known that this algorithm is equivalent to computing a noncommutative Gröbner Basis for  $R$ . This article shows that we can also use noncommutative Involutive Basis methods to obtain an involutive complete rewrite system for  $R$ .

## 1 Introduction

Let  $R = \langle A \mid B \rangle$  be a monoid rewrite system, where  $A = \{a_1, \dots, a_n\}$  is an alphabet and  $B = \{b_1, \dots, b_m\}$  is a set of rules of the form  $b_i = \ell_i \rightarrow r_i$  ( $1 \leq i \leq m$ ;  $\ell_i, r_i \in A^*$ ). Given a fixed admissible well-order on the words in  $A$  compatible with  $R$ , the Knuth-Bendix critical pairs completion algorithm [1] attempts to find a complete rewrite system  $R'$  for  $R$  that is Noetherian and confluent, so that any word over the alphabet  $A$  has a unique normal form with respect to  $R'$ . The algorithm proceeds by considering overlaps of left hand sides of rules, forming new rules when two reductions of an overlap word result in two distinct normal forms.

It is well known (see for example [2]) that the Knuth-Bendix critical pairs completion algorithm is a special case of the noncommutative Gröbner Basis algorithm [3]. To find the complete rewrite system, we treat the rewrite system  $R$  as a set of polynomials  $F = \{\ell_1 - r_1, \ell_2 - r_2, \dots, \ell_m - r_m\}$  generating a two-sided ideal over the noncommutative polynomial ring  $\mathbb{Z}\langle a_1, \dots, a_n \rangle$ , and compute a noncommutative Gröbner Basis  $G$  for  $F$  using a monomial ordering induced from the fixed admissible well-order on the words in  $A$ . This is done by computing an S-polynomial for each overlap of lead monomials, a polynomial which is added to the system if it does not reduce to zero.

*Example 1.* Consider the monoid rewrite system  $R = \langle x, y \mid xy^2 \rightarrow \varepsilon, x^2 \rightarrow \varepsilon \rangle$ , where  $\varepsilon$  denotes the empty word. Let the ordering on words be DegLex, so that  $w_1 > w_2$  if  $\deg(w_1) > \deg(w_2)$  or, in the case that the degrees are equal, the first (say  $i$ -th) letter (working left-to-right) on which the words  $w_1$  and  $w_2$  differ is such that the  $i$ -th letter of  $w_1$  is greater than the  $i$ -th letter of  $w_2$  in the ordering given to the alphabet (in this example, we will take  $x > y$ ).

In the first pass of the Knuth-Bendix algorithm, the only overlap of left hand sides that yields a new rule is the overlap shown below.

$$\begin{array}{ccc}
 & x(xy^2) = x^2y^2 = (x^2)y^2 & \\
 & \swarrow \quad \searrow & \\
 xy^2 \rightarrow \varepsilon & & x^2 \rightarrow \varepsilon \\
 \swarrow & \longleftarrow & \searrow \\
 x & & y^2
 \end{array}$$

This corresponds to the S-polynomial

$$x(xy^2 - 1) - (x^2 - 1)y^2 = y^2 - x$$

in the Gröbner Basis algorithm. After all overlaps or S-polynomials have been considered, both algorithms terminate with the complete rewrite system  $R' = \langle x, y \mid x^2 \rightarrow \varepsilon, y^2 \rightarrow x, xy \rightarrow yx \rangle$  for  $R$ .

In [4], an alternative method of computing noncommutative Gröbner Bases was proposed, using concepts from the theory of commutative Involutive Bases [5]. We shall now apply these ideas to monoid rewrite systems, giving a third method of obtaining a complete rewrite system.

## 2 Involutive Rewrite Systems

In any monoid rewrite system, we can apply a rule  $\ell \rightarrow r$  to a word  $\omega$  if and only if  $\omega$  has the form  $\omega = ulv$ , where  $u, v$  may be the empty word. In this situation, we say that the word  $\omega$  reduces to the word  $\omega' = urv$ . In an involutive monoid rewrite system, we restrict these reductions by introducing additional properties each reduction must satisfy, based on an assignment of left and right *multiplicative letters* to  $\ell$ .

### 2.1 Involutive Reductions

In order to determine which letters are multiplicative for a particular left hand side of a rewrite rule, we choose an *involutive reduction*  $I$  which, given any set of words  $W$  over an alphabet  $A = \{a_1, \dots, a_n\}$ , assigns a set of left multiplicative letters  $\mathcal{M}_I^L(w, W) \subseteq \{a_1, \dots, a_n\}$  and a set of right multiplicative letters  $\mathcal{M}_I^R(w, W) \subseteq \{a_1, \dots, a_n\}$  to any word  $w \in W$ . The multiplicative letters for a set of rewrite rules are then determined by the assignment of multiplicative letters to the set of left hand sides of the rewrite rules.

**Definition 1.** (a) A word  $\ell$  is an *involutive divisor* of a word  $\omega$  with respect to some involutive reduction  $I$ , written  $\ell \mid_I \omega$ , if  $\omega = ulv$  for some words  $u, v$ ; the suffix of  $u$  of degree 1 (if it exists) is left multiplicative for  $\ell$ ; and the prefix of  $v$  of degree 1 (again if it exists) is right multiplicative for  $\ell$ .

(b) A rule  $b = \ell \rightarrow r$  is an *involutive divisor* of a word  $\omega$  with respect to some involutive reduction  $I$ , written  $b \mid_I \omega$ , if  $\ell \mid_I \omega$ . Further, if  $\omega = ulv$ , we say that the word  $\omega$  involutively reduces to the word  $\omega' = urv$  using the rule  $b$ .

*Example 2.* Let  $W = \{xy, yz\}$  and  $A = \{x, y, z\}$ . Define left and right multiplicative letters as follows.

$w$	$\mathcal{M}_I^L(w, W)$	$\mathcal{M}_I^R(w, W)$
$xy$	$\{x, y, z\}$	$\{y, z\}$
$yz$	$\{y, z\}$	$\{x\}$

We consider reductions of  $\omega = xyz$  given the set of rules  $B = \{xy \rightarrow z, yz \rightarrow x\}$ . Conventionally, we are able to reduce  $\omega$  by both of the rules in  $B$ , giving reductions  $z^2$  and  $x^2$  respectively. Involutively, we see that the first rule in  $B$  reduces  $\omega$  (because  $\omega = (xy)z$  and the letter  $z$  is right multiplicative for the word  $xy$ ); but the second rule in  $B$  does not reduce  $\omega$  (because  $\omega = x(yz)$  and the letter  $x$  is not left multiplicative for  $yz$ ).

There are many ways of defining an involutive reduction, but we will only be interested in involutive reductions that will enable us to find complete rewrite systems. One such family of reductions is the family of *strong* reductions.

**Definition 2.** Let  $A^*$  be the set of all words over the alphabet  $A = \{a_1, \dots, a_n\}$ , and let  $W \subset A^*$ . The involutive cone  $\mathcal{C}_I(w, W)$  of any word  $w \in W$  with respect to some involutive reduction  $I$  is defined as follows:

$$\begin{aligned} \mathcal{C}_I^L(w, W) &= \{u \in A^* \text{ such that } w \mid_I uw\}; \\ \mathcal{C}_I^R(w, W) &= \{v \in A^* \text{ such that } w \mid_I wv\}; \\ \mathcal{C}_I(w, W) &= \{uvw \text{ such that } u \in \mathcal{C}_I^L(w, W), v \in \mathcal{C}_I^R(w, W)\}. \end{aligned}$$

**Definition 3.** A strong involutive reduction  $I$  is defined on  $A^*$  if we can assign a set of left multiplicative letters  $\mathcal{M}_I^L(w, W)$  and a set of right multiplicative letters  $\mathcal{M}_I^R(w, W)$  to any word  $w$  in any set of words  $W \subset A^*$  such that the following three conditions are satisfied.

- If there exist two words  $w_1, w_2 \in W$  such that  $\mathcal{C}_I(w_1, W) \cap \mathcal{C}_I(w_2, W) \neq \emptyset$ , then either  $\mathcal{C}_I(w_1, W) \subset \mathcal{C}_I(w_2, W)$  or  $\mathcal{C}_I(w_2, W) \subset \mathcal{C}_I(w_1, W)$ .
- Any word  $\omega \in \mathcal{C}_I(w, W)$  is involutively reducible by  $w$  in one way only, so that if  $w$  appears as a subword of  $\omega$  in more than one way, then only one of these ways allows us to deduce that  $w$  involutively reduces  $\omega$ .
- If  $W' \subset W$ , then for all  $w' \in W'$ ,  $\mathcal{M}_I^L(w', W) \subseteq \mathcal{M}_I^L(w', W')$  and  $\mathcal{M}_I^R(w', W) \subseteq \mathcal{M}_I^R(w', W')$ .

*Remark 1.* We shall refer to the three conditions of Definition 3 as (respectively) the Disjoint Cones condition, the Unique Divisor condition and the Subset condition.

**Definition 4.** If an involutive reduction determines the left and right multiplicative letters for a word  $w \in W$  independently of the set  $W$ , then the reduction is known as a global involutive reduction. Otherwise, the reduction is known as a local involutive reduction.

---

**Algorithm 1** An Involutive Normal Form Algorithm
 

---

**Inputs:** A rule  $b = \ell \rightarrow r$  and a set of rules  $B = \{\ell_1 \rightarrow r_1, \dots, \ell_m \rightarrow r_m\}$  over an alphabet  $A = \{a_1, \dots, a_n\}$ ; a fixed admissible well-order  $<$  on words in  $A$  compatible with  $R = \langle A \mid B \rangle$ ; an involutive reduction  $I$ .

**Output:**  $\text{INF}_I(b, B)$ , the involutive normal form  $\lambda \rightarrow \rho$  of  $b$  with respect to  $B$  and  $I$ .

```

 $\lambda = \varepsilon; \rho = \varepsilon; \text{first} = \text{true};$ 
while ( $\ell \neq \varepsilon$ ) do
   $j = 1; \text{found} = \text{false};$ 
  while ( $j \leq m$ ) and ( $\text{found} == \text{false}$ ) do
    if ( $\ell = u\ell_j v$  for some words  $u, v$  such that  $\ell_j \mid_I \ell$ ) then
       $\text{found} = \text{true};$ 
       $\ell = ur_j v$  (if there are several candidates for  $u$  (and therefore for  $v$ ), choose
      the one with the smallest degree);
      if ( $\ell < r$ ) then
        swap  $\ell$  and  $r$ ;
      else if ( $\ell == r$ ) and ( $\text{first} == \text{true}$ ) then
        return  $\varepsilon \rightarrow \varepsilon$  (the empty rule);
      end if
    else
       $j = j + 1;$ 
    end if
  end while
  if ( $\text{found} == \text{false}$ ) then
    if ( $\text{first} == \text{true}$ ) then
       $\lambda = \ell; \ell = r; r = \varepsilon; \text{first} = \text{false};$ 
    else
       $\rho = \ell; \ell = \varepsilon;$ 
    end if
  end if
end while
return  $\lambda \rightarrow \rho;$ 

```

---

## 2.2 Constructing Involutive Complete Rewrite Systems

Whereas the Knuth-Bendix algorithm constructs a complete rewrite system by considering overlaps of left hand sides of rewrite rules, and the noncommutative Gröbner Basis algorithm constructs a complete rewrite system by considering S-polynomials, the Involutive Basis algorithm for monoid rewrite systems uses processes known as *prolongation* and *autoreduction*.

**Definition 5.** Given a rule  $b = \ell \rightarrow r$ , a left prolongation of  $b$  is a rule  $a_i \ell \rightarrow a_i r$ , where  $a_i$  is a left nonmultiplicative letter of  $\ell$  with respect to some involutive reduction  $I$ ; and a right prolongation of  $b$  is a rule  $\ell a_j \rightarrow r a_j$ , where  $a_j$  is a right nonmultiplicative letter of  $\ell$  with respect to  $I$ .

**Definition 6.** A set of rules  $B$  is said to be autoreduced if no rule  $b \in B$  exists such that  $b$  is involutively reducible (with respect to  $B$ ) by some rule  $b' \in B \setminus \{b\}$ .

*Remark 2.* In the algorithm to perform autoreduction (Algorithm 2), we use the following notation: let  $\text{INF}_I(b, B, B')$  denote the involutive normal form of the rule  $b$  with respect to the set of rules  $B$  (obtained using Algorithm 1), where reductions (in Algorithm 1) are only to be performed by elements of the set  $B' \subseteq B$ .

---

### Algorithm 2 Autoreduction

---

**Inputs:** A set of rules  $B = \{\ell_1 \rightarrow r_1, \dots, \ell_m \rightarrow r_m\}$ ; an involutive reduction  $I$ .

**Output:** An autoreduced set of rules  $B' = \{\ell'_1 \rightarrow r'_1, \dots, \ell'_{m'} \rightarrow r'_{m'}\}$ .

```

while ( $\exists b_i \in B$  such that  $\text{INF}_I(b_i, B, B \setminus \{b_i\}) \neq b_i$ ) do
   $b'_i = \text{INF}_I(b_i, B, B \setminus \{b_i\})$ ;
   $B = B \setminus \{b_i\}$ ;
  if ( $b'_i \neq \varepsilon \rightarrow \varepsilon$ ) then
     $B = B \cup \{b'_i\}$ ;
  end if
end while
 $B' = B$ ;
return  $B'$ ;

```

---

*Remark 3.* With respect to a strong involutive reduction, the involutive cones of an autoreduced set of rules are always disjoint.

In the following definitions, let  $R$  be a rewrite system and let  $I$  be a fixed involutive reduction.

**Definition 7.**  $R$  is a locally involutive rewrite system if  $R$  is autoreduced and all prolongations of rules in  $R$  involutively reduce to the empty rule using  $R$ .

**Definition 8.**  $R$  is an involutive rewrite system if  $R$  is autoreduced and all possible multiples  $ulv \rightarrow urv$  of rules  $\ell \rightarrow r$  in  $R$  involutively reduce to the empty rule using  $R$  ( $u$  and  $v$  are any words).

Consider a word  $w$  of degree  $d$ .

- Let  $\text{Prefix}(w, i)$  denote the prefix of  $w$  of degree  $i$  (where  $1 \leq i \leq d$ ).
- Let  $\text{Suffix}(w, i)$  denote the suffix of  $w$  of degree  $i$  (where  $1 \leq i \leq d$ ).
- Let  $\text{Subword}(w, i, j)$  denote the subword of  $w$  starting at position  $i$  and finishing at position  $j$  (where  $1 \leq i \leq j \leq d$ ).

**Definition 9.** Let  $I$  be a fixed involutive reduction; let  $\omega$  be a fixed word; let  $W$  be any set of words; and consider any sequence  $(w_1, w_2, \dots, w_k)$  of words from  $W$  ( $w_i \in W$  for all  $1 \leq i \leq k$ ), each of which is a subword of  $\omega$  (so that  $\omega = u_i w_i v_i$  for all  $1 \leq i \leq k$ , where the  $u_i$  and the  $v_i$  are words). For all  $1 \leq i < k$ , suppose that the word  $w_{i+1}$  satisfies exactly one of the following conditions.

- (a)  $w_{i+1}$  involutively reduces a left prolongation of  $w_i$ , so that  $\deg(u_i) \geq 1$ ;  $\text{Suffix}(u_i, 1) \notin \mathcal{M}_I^L(w_i, W)$ ; and  $w_{i+1} \mid_I (\text{Suffix}(u_i, 1))w_i$ .
- (b)  $w_{i+1}$  involutively reduces a right prolongation of  $w_i$ , so that  $\deg(v_i) \geq 1$ ;  $\text{Prefix}(v_i, 1) \notin \mathcal{M}_I^R(w_i, W)$ ; and  $w_{i+1} \mid_I w_i (\text{Prefix}(v_i, 1))$ .

Then  $I$  is continuous at  $\omega$  if all the pairs  $(u_i, v_i)$  are disjoint ( $(u_i, v_i) \neq (u_j, v_j)$  for all  $i \neq j$ );  $I$  is a continuous involutive reduction if  $I$  is continuous for all possible  $\omega$ .

**Proposition 1.** If an involutive reduction  $I$  is continuous; and if an arbitrary set of rules  $B$  forms part of a locally involutive rewrite system  $R$  with respect to  $I$  and some fixed admissible well-order  $<$  on words compatible with  $R$ , then  $R$  is an involutive rewrite system with respect to  $I$  and  $<$ .

*Proof.* Assume that  $B = \{\ell_1 \rightarrow r_1, \dots, \ell_m \rightarrow r_m\}$ , and let  $L = \{\ell_1, \dots, \ell_m\}$  denote the set of left hand sides. Given any rule  $b_i \in B$  and any words  $u$  and  $v$ , in order to show that  $R$  is an involutive rewrite system with respect to  $I$  and  $<$ , we must show that  $ub_i v$  involutively reduces to the empty rule using  $B$ .

If  $\ell_i \mid_I ul_i v$  we are done, as we can use  $b_i$  to involutively reduce  $ub_i v$  to obtain the empty rule. Otherwise, either  $\exists a_1 \notin \mathcal{M}_I^L(\ell_i, L)$  such that  $a_1 = \text{Suffix}(u, 1)$ , or  $\exists a_1 \notin \mathcal{M}_I^R(\ell_i, L)$  such that  $a_1 = \text{Prefix}(v, 1)$ . Without loss of generality, assume that the first case applies. By local involutivity, the prolongation  $a_1 b_i$  involutively reduces to the empty rule using  $B$ . Assuming that the first step of this involutive reduction involves the rule  $b_j \in B$ , we can write

$$a_1 b_i = a_1 \ell_i \rightarrow a_1 r_i = u_1 \ell_j v_1 \rightarrow a_1 r_i, \quad (1)$$

where  $u_1, v_1$  are words such that  $\ell_j \mid_I a_1 \ell_i$ . Multiplying both sides of Equation (1) on the left by  $u' := \text{Prefix}(u, \deg(u) - 1)$  and on the right by  $v$ , we obtain the equation

$$ub_i v = u' u_1 \ell_j v_1 v \rightarrow ur_i v. \quad (2)$$

If  $\ell_j \mid_I ul_i v$ , it is clear that we can use  $b_j$  to involutively reduce the rule  $ub_i v$  to obtain the rule  $u' u_1 r_j v_1 v \rightarrow ur_i v$  or  $ur_i v \rightarrow u' u_1 r_j v_1 v$  (dependent upon whether  $u' u_1 r_j v_1 v > ur_i v$  or not). We then continue by induction, noticing (i) that the left hand side of our new rule will contain the left hand side of some rule in  $B$  as a subword because we know that the prolongation  $a_1 b_i$  involutively reduces to the empty rule; and (ii) the process will terminate because of the admissibility of our word ordering.

Otherwise, if  $b_j$  does not involutively reduce  $ub_i v$ , either  $\exists a_2 \notin \mathcal{M}_I^L(\ell_j, L)$  such that  $a_2 = \text{Suffix}(u' u_1, 1)$ , or  $\exists a_2 \notin \mathcal{M}_I^R(\ell_j, L)$  such that  $a_2 = \text{Prefix}(v_1 v, 1)$ . This time (again without loss of generality), assume that the second case applies. By local involutivity, the prolongation  $b_j a_2$  involutively reduces to the empty rule using  $B$ . Assuming that the first step of this involutive reduction involves the rule  $b_k \in B$ , we can write

$$b_j a_2 = \ell_j a_2 \rightarrow r_j a_2 = u_2 \ell_k v_2 \rightarrow r_j a_2, \quad (3)$$

where  $u_2, v_2$  are words such that  $\ell_k \mid_I \ell_j a_2$ . Multiplying both sides of Equation (3) on the left by  $u' u_1$  and on the right by  $v' := \text{Suffix}(v_1 v, \deg(v_1 v) - 1)$ , we obtain the equation

$$u' u_1 b_j v_1 v = u' u_1 u_2 \ell_k v_2 v' \rightarrow u' u_1 r_j v_1 v \quad (4)$$

and hence

$$ub_i v = u' u_1 u_2 \ell_k v_2 v' \rightarrow ur_i v. \quad (5)$$

If  $\ell_k \mid_I ul_i v$ , the proof (as before) is now complete. Otherwise, we continue by induction, obtaining a sequence  $b_i, b_j, b_k, \dots$  of elements in  $B$ . By construction, the left hand side of each rule in the sequence is a subword of  $ul_i v$ . By continuity (at  $ul_i v$ ), no two elements in the sequence reduce  $ul_i v$  in the same way. Because  $ul_i v$  has a finite number of subwords, the sequence must be finite, terminating with a rule  $b' \in B$  such that  $b' \mid_I ul_i v$ . This then allows us to finish the proof through use of the local involutivity of  $B$  and the admissibility of our word ordering.  $\square$

Let us now consider Algorithm 3 (on page 8), an algorithm to compute a locally involutive rewrite system for an input rewrite system  $R = \langle A \mid B \rangle$ . The algorithm starts by autoreducing  $B$  using Algorithm 2. We then construct a set  $S$  containing all the possible prolongations of elements of  $B$ , before recursively (a) picking a prolongation  $s$  from  $S$  such that the left hand side of  $s$  is minimal with respect to the chosen word ordering; (b) removing  $s$  from  $S$ ; and (c) finding the involutive normal form  $s'$  of  $s$  with respect to  $B$ . If during this loop a normal form  $s'$  is found that is nonzero, we exit the loop and autoreduce the set  $B \cup \{s'\}$ , continuing thereafter to construct a new set  $S$  and repeating the above process

---

**Algorithm 3** The Involutive Basis Algorithm for Monoid Rewrite Systems
 

---

**Inputs:** A rewrite system  $R = \langle A \mid B \rangle$ , where  $A = \{a_1, \dots, a_n\}$  is an alphabet and  $B = \{\ell_1 \rightarrow r_1, \dots, \ell_m \rightarrow r_m\}$  is a set of rules; a fixed admissible well-order  $<$  on words in  $A$  compatible with  $R$ ; an involutive reduction  $I$ .

**Output:** (In the case of termination): A locally involutive rewrite system  $R'$  for  $R$ .

```

 $C = \emptyset;$ 
 $B = \text{Autoreduce}(B);$ 
while ( $C == \emptyset$ ) do
   $S = \{a_i b \mid b \in B, a_i \notin \mathcal{M}_I^L(b, B)\} \cup \{b a_i \mid b \in B, a_i \notin \mathcal{M}_I^R(b, B)\};$ 
   $s' = 0;$ 
  while ( $S \neq \emptyset$ ) and ( $s' == 0$ ) do
    Let  $s$  be a rule in  $S$  whose left hand side is minimal with respect to  $<;$ 
     $S = S \setminus \{s\};$ 
     $s' = \text{INF}_I(s, B);$ 
  end while
  if ( $s' \neq \varepsilon \rightarrow \varepsilon$ ) then
     $B = \text{Autoreduce}(B \cup \{s'\});$ 
  else
     $C = B;$ 
  end if
end while
return  $R' = \langle A \mid C \rangle;$ 

```

---

on this new set. If however all the prolongations in  $S$  involutively reduce to the empty rule, then by definition  $R' = \langle A \mid B \rangle$  is a locally involutive rewrite system, and so we can exit the algorithm with this rewrite system.

**Definition 10.**  *$R$  is a locally involutive complete rewrite system if  $R$  is (i) a complete rewrite system; and (ii) a locally involutive rewrite system with respect to some involutive reduction  $I$ .*

**Definition 11.**  *$R$  is an involutive complete rewrite system if  $R$  is (i) a complete rewrite system; and (ii) an involutive rewrite system with respect to some involutive reduction  $I$ .*

**Proposition 2.** *If  $I$  is a strong involutive reduction, then any involutive rewrite system  $R$  with respect to  $I$  is an involutive complete rewrite system with respect to  $I$ .*

*Proof.* By definition of a strong reduction, the involutive normal form of any word  $\omega$  with respect to  $R$  is unique. We can therefore show that  $R$  is an involutive complete rewrite system by showing that the conventional and involutive normal forms of an arbitrary word  $\omega$  with respect to  $R$  are identical. For this it is sufficient to show that a word  $\omega$  is conventionally reducible by  $R$  if and only if it is involutively reducible by  $R$ . ( $\Rightarrow$ ) Trivial as every involutive reduction is a conventional reduction. ( $\Leftarrow$ ) Let  $b = \ell \rightarrow r$  be an arbitrary rule in  $R$ . If a word  $\omega$  is conventionally reducible by  $b$ , it follows that  $\omega = ulv$  for some words  $u$  and  $v$ . But  $R$  is an involutive rewrite system, so there must exist a rule  $b' \in R$  such that  $b' \mid_I ulv$ . Thus  $\omega$  is also involutively reducible by  $R$ .  $\square$

### 3 A Global Involutive Reduction

**Definition 12 (The Left Reduction).** *Given any word  $w$ , the left reduction  $\triangleleft$  assigns all letters to be left multiplicative for  $w$ , and assigns all letters to be right nonmultiplicative for  $w$ .*

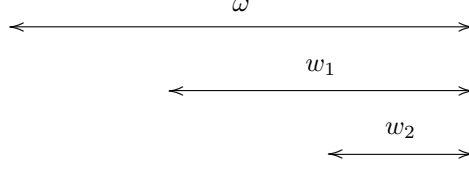
**Proposition 3.** *The left reduction is a strong involutive reduction.*

*Proof.* We need to show that the three conditions of Definition 3 hold.

#### – Disjoint Cones Condition

Consider two involutive cones  $\mathcal{C}_{\triangleleft}(w_1)$  and  $\mathcal{C}_{\triangleleft}(w_2)$  associated to two words  $w_1, w_2$  over some alphabet  $A$ . If  $\mathcal{C}_{\triangleleft}(w_1) \cap \mathcal{C}_{\triangleleft}(w_2) \neq \emptyset$ , then there must be some word  $\omega \in A^*$  such that  $\omega$  contains both words  $w_1$  and  $w_2$  as subwords, and (as placed in  $\omega$ ) both  $w_1$  and  $w_2$  must involutively reduce  $\omega$ . By definition of  $\triangleleft$ , both  $w_1$  and  $w_2$  must be suffices of  $\omega$ . Thus, assuming (without loss of generality) that  $\deg(w_1) > \deg(w_2)$ , we are able to draw the following

diagram summarising the situation.



But now, assuming that  $w_1 = w_3w_2$  for some word  $w_3$ , it is clear that  $\mathcal{C}_{\triangleleft}(w_1) \subset \mathcal{C}_{\triangleleft}(w_2)$  because any word  $\omega' \in \mathcal{C}_{\triangleleft}(w_1)$  must be of the form  $\omega' = w'w_1$  for some word  $w'$ ; this means that  $\omega' = w'w_3w_2 \in \mathcal{C}_{\triangleleft}(w_2)$ .

– **Unique Divisor Condition**

As a word  $\omega$  is only involutively reducible by a word  $w$  with respect to the left reduction if  $w$  is a suffix of  $\omega$ , it is clear that  $w$  can only involutively reduce  $\omega$  in at most one way.

– **Subset Condition**

Follows immediately due to the left reduction being a global reduction.

□

**Proposition 4.** *The left reduction is continuous.*

*Proof.* Let  $\omega$  be an arbitrary fixed word; let  $W$  be any set of words; and consider any sequence  $(w_1, w_2, \dots, w_k)$  of words from  $W$  ( $w_i \in W$  for all  $1 \leq i \leq k$ ), each of which is a subword of  $\omega$  (so that  $\omega = u_iw_iv_i$  for all  $1 \leq i \leq k$ , where the  $u_i$  and the  $v_i$  are words). For all  $1 \leq i < k$ , suppose that the word  $w_{i+1}$  satisfies condition (b) of Definition 9 (condition (a) can never be satisfied because  $\triangleleft$  never assigns any left nonmultiplicative letters). To show that  $\triangleleft$  is continuous, we must show that no two pairs  $(u_i, v_i)$  and  $(u_j, v_j)$  are the same, where  $i \neq j$ .

Consider an arbitrary word  $w_i$  from the sequence, where  $1 \leq i < k$ . Because  $\triangleleft$  assigns no right multiplicative letters, the next word  $w_{i+1}$  in the sequence must be a suffix of the prolongation  $w_i(\text{Prefix}(v_i, 1))$  of  $w_i$ , so that  $\deg(v_{i+1}) = \deg(v_i) - 1$ . It is therefore clear that no two identical  $(u, v)$  pairs can be found in the sequence, as  $\deg(v_1) > \deg(v_2) > \dots > \deg(v_k)$ . □

*Remark 4.* Now that we know that the left reduction is strong and continuous, we can state that (with respect to the left reduction) any locally involutive rewrite system returned by Algorithm 3 is an involutive complete rewrite system.

### 3.1 A Group Example: $S_3$

*Example 3.* Let  $R = \langle Y, X, y, x \mid x^3 \rightarrow \varepsilon, y^2 \rightarrow \varepsilon, (xy)^2 \rightarrow \varepsilon, Xx \rightarrow \varepsilon, xX \rightarrow \varepsilon, Yy \rightarrow \varepsilon, yY \rightarrow \varepsilon \rangle$  be a monoid rewrite system for the group  $S_3$ , where  $Y > X > y > x$  is the ordering given to the alphabet. (Note: this presentation is obtained from the group presentation  $S_3 = \langle y, x \mid x^3, y^2, (xy)^2 \rangle$ ;  $X$  and  $Y$  represent the inverses of  $x$  and  $y$  respectively.)

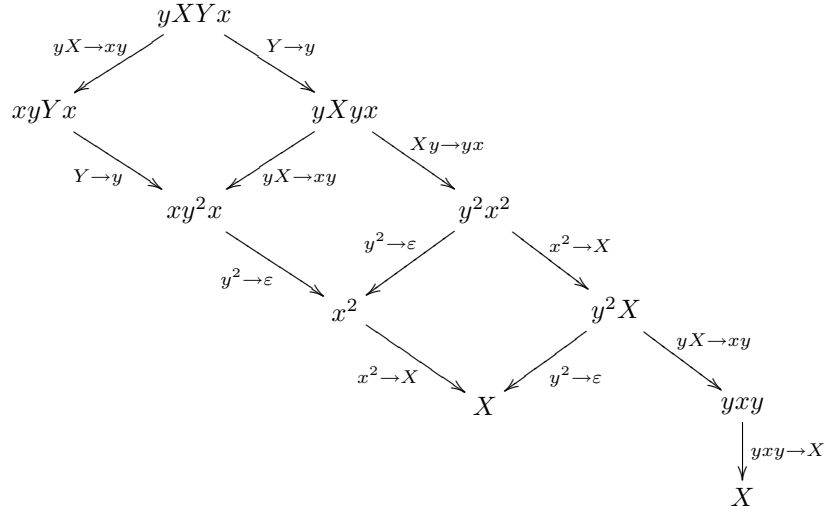
If we apply the Knuth-Bendix algorithm to  $R$  with respect to the DegLex word ordering, we obtain the complete rewrite system

$$R' = \langle Y, X, y, x \mid xyx \rightarrow y, yxy \rightarrow X, x^2 \rightarrow X, Xx \rightarrow \varepsilon, y^2 \rightarrow \varepsilon, Xy \rightarrow yx, xX \rightarrow \varepsilon, yX \rightarrow xy, X^2 \rightarrow x, Y \rightarrow y \rangle.$$

With respect to the left reduction, if we apply Algorithm 3 to  $R$ , we obtain the involutive complete rewrite system

$$R'' = \langle Y, X, y, x \mid y^2 \rightarrow \varepsilon, Xx \rightarrow \varepsilon, xX \rightarrow \varepsilon, Yy \rightarrow \varepsilon, y^2x \rightarrow x, Y \rightarrow y, Yx \rightarrow yx, Xxy \rightarrow y, Yyx \rightarrow x, x^2 \rightarrow X, X^2 \rightarrow x, xyx \rightarrow y, Xy \rightarrow yx, Xyx \rightarrow xy, x^2y \rightarrow yx, yX \rightarrow xy, yxy \rightarrow X, Yxy \rightarrow X, YX \rightarrow xy \rangle.$$

With the involutive complete rewrite system, we are now able to uniquely reduce each word over the alphabet  $\{Y, X, y, x\}$  to one of the six elements of  $S_3$ . To illustrate this, consider the word  $yXYx$ . Using the 10 element complete rewrite system  $R'$  obtained by using the Knuth-Bendix algorithm, there are several reduction paths for this word, as illustrated by the following diagram.



However, by involutively reducing the word  $yXYx$  with respect to the 19 element involutive complete rewrite system  $R''$ , there is only one reduction path, namely

$$\begin{array}{c} yXYx \\ \downarrow Yx \rightarrow yx \\ yXyx \\ \downarrow Xyx \rightarrow xy \\ yxy \\ \downarrow yxy \rightarrow X \\ X \end{array}$$

## 4 Conclusions and Further Work

We have demonstrated an alternative way of obtaining a complete rewrite system for a monoid rewrite system, one in which unique normal forms are also obtained uniquely. Although the correctness of the algorithm used to compute such complete rewrite systems was shown, the question of termination still remains, in particular the question of termination in the case that the Knuth-Bendix critical pairs completion algorithm terminates. To answer this question, further investigation is needed, which may involve more complicated involutive reductions such as those introduced in [6]. It would also be interesting to investigate the efficiency of Algorithm 3 compared to the traditional methods; research in [7] shows that for certain examples in the theory of commutative involutive bases, the involutive method is more efficient than the traditional Gröbner Basis approach.

## 5 Acknowledgements

The author would like to thank Prof. Larry Lambe for providing the C libraries in which the algorithms described in this article have been implemented in. This work was supported by the EPSRC.

## References

1. Knuth, D.E., Bendix, P.B.: Simple word problems in universal algebras. In Leech, J., ed.: *Computational Problems in Abstract Algebra*, Pergamon Press (1970) 263–297
2. Heyworth, A.: Rewriting as a special case of Gröbner Basis Theory. In Atkinson, M., Gilbert, N.D., Howie, J., Linton, S.A., Robertson, E.F., eds.: *Computational and Geometric Aspects of Modern Algebra* (Edinburgh, 1998). Volume 275 of *Proc. London Math. Soc.* Cambridge University Press (2000) 101–105
3. Mora, T.: Gröbner Bases for non-commutative polynomial rings. In Calmet, J., ed.: *AAECC-3: Proc. 3rd Int. Conf. on Algebraic Algorithms and Error-Correcting Codes* (Grenoble, France, July 15–19, 1985). Volume 223 of *Lecture Notes in Comput. Sci.* Springer (1986) 353–362
4. Evans, G.A.: Noncommutative Involutive Bases. In Tran, Q.N., ed.: *Proc. 10th Int. Conf. Applications of Computer Algebra*, Beaumont, Texas, USA (2004) 49–64
5. Zharkov, A.Yu., Blinkov, Yu.A.: Involution Approach to Investigating Polynomial Systems. *Math. Comput. Simulation* **42** (1996) 323–332
6. Evans, G.A.: Noncommutative Involutive Bases. (PhD Thesis, University of Wales, Bangor). To appear, 2006.
7. Gerdt, V.P., Blinkov, Yu.A.: Involutive Bases of Polynomial Ideals. *Math. Comput. Simulation* **45** (1998) 519–541