# The Unification Problem for One Relation Thue Systems*

Christopher Lynch

Department of Mathematics and Computer Science Box 5815, Clarkson University, Potsdam, NY 13699-5815, USA,
clynch@sun.mcs.clarkson.edu,
http://www.clarkson.edu/∼clynch

**Abstract.** We give an algorithm for the unification problem for a generalization of Thue Systems with one relation. The word problem is a special case. We show that in many cases this is a decision procedure with at most an exponential time bound. We conjecture that this is always a decision procedure.

## 1 Introduction

In this paper we study the unification problem and word problem for Thue Systems. This basic problem appears under several different names. It is also known as the unification and word problem for semigroups, terms with monadic function symbols, and ground terms with one associative operator.

In particular, we are interested in Thue Systems with only one equation, but we have generalized our results to larger classes. The word problem for one equation can be stated simply: Given an equation $s = t$, and two words $u_0$ and $u_n$, is there a sequence of words $u_0, \cdots, u_n$ such that each $u_{i+1}$ is the result of replacing an occurrence of $s$ in $u_i$ by $t$, or replacing an occurrence of $t$ in $u_i$ by $s$?

Despite the very simple formulation of the problem, it is unknown whether the problem is decidable. It has been shown to be undecidable when there are three equations instead of one [9], but the case for two equations is also unknown. The word problem for groups with one defining equation has been known to be decidable for 65 years [8]. However, despite considerable work in the area [2,4] (see [6] for a survey), the decidability for one equation Thue systems is unknown.

In this paper, we address (but do not solve) the problem, and we also generalize the problem in some ways. One of our generalizations is to consider the unification problem, which is a generalization of the word problem. The unification problem is as follows: Given an equation $s = t$ and words $u$ and $v$, are there words $x$ and $y$ such that $vy$ can be reached from $ux$ with a sequence of replacements of $s$ by $t$ and $t$ by $s$. We also generalize from one equation Thue systems to allow more than one equation but require a certain syntactic structure. Our

* This work was supported by NSF grant number CCR-9712388 and partially done while visiting INRIA Lorraine and CRIN.

Jacques Calmet and Jan Plaza (Eds.): AISC'98, LNAI 1476, pp. 195–208, 1998.

result is a procedure that decides the unification problem when it halts, and also produces the most general unifier. We have not been able to prove that it halts for all instances of our generalization of the one equation unification problem, but we conjecture that it does.

Although we have not proved a decidability result, we believe our work is important. We have provided some theorems showing how to automatically detect that the algorithm is a decision procedure for certain Thue systems. We even give a complexity result, showing that the algorithm is at most exponential for a large class of Thue systems We have implemented an algorithm for one equation Thue systems, based on the one in this paper. On every example we have tried, it always terminates quickly with the answer.

Our main interest in this problem is not just for one equation Thue systems. Our goal is to extend these results to equations over terms. Popular methods for deciding word and unification problems, like the Knuth-Bendix completion method have many examples, even very simple ones, where they do not halt. Our method attempts to avoid those problems. Although our presentation here is only for monadic function symbols, the ideas extend to function symbols of higher arity. The syntactic restrictions on the class used in this paper allow for our algorithm to be deterministic. Relaxing those restrictions is possible if we allow the algorithm to be non-deterministic. Our plan for the near future is to investigate all these extensions. We expect the ideas in this paper will be important for finding decision procedures for interesting classes of equational theories. The main inspiration for our paper is our previous work on SOUR graphs [7]. This paper is actually a simplification of those ideas, although the ideas have evolved quite a lot. We have achieved the two main purposes we sought in the evolution of those ideas: First, they are vastly simplified to allow much easier understanding and implementation. Second, we have shown the use of the method to solve decision problems, which we did not realize before.

The next section of the paper gives some required background. The section after that builds up the necessary machinery for our algorithm. We convert the unification problem into a problem in rewrite systems. The following section develops the rewrite system problem into an algorithm. Interestingly, in this section we show that the unification (and word) problem is equivalent to a problem in termination of rewrite systems. We show how to detect loops in rewrite systems, and conjecture that all nonterminating rewrite sequences are loops, which forms the basis of our algorithm. Interesting this is the same conjecture made for termination of one rule semi-Thue systems [10], another decision problem whose solution is unknown. That gives us the impression that the same techniques used for solving the termination problem will be useful for solving the word (and unification) problem. In the conclusion, we relate our work with other work.

## 2 Preliminaries

We are given a set $A$ as *alphabet.* In this paper, we use letters $a, b, c, d, e, f, g, h$ as members of the alphabet. A *word* is a sequence of members of the alphabet. We use letters $r, s, t, u, v, w$ to represent words. If $w$ is a word then $|w|$ is the number of symbols in $w$. If $|w| = 0$, then we write $w$ as $\epsilon$ and call it the *empty word.* If $u$ and $v$ are words, then $uv$ represents the concatenation of $u$ and $v$. Then $u$ is a *prefix* of $uv$, and $v$ is a suffix of $uv$. Also, $v$ is a subword of $uvw$. $u \approx v$ is an *equation* if $u \neq \epsilon$ and $v \neq \epsilon$.[1]

A *Thue System* is a set of equations. We assume it is closed under symmetry.[2] Let $E$ be a Thue system. If $s \approx t \in E$ then we write $usv \approx_E utv$. If $E$ is obvious, we may write $usv \approx utv$. We call this an *equational step at position* $p$, where $p = |u|$. If $p = 0$, then this is called an *equational step at the top.* A *proof of* $u \approx_E v$ is of the form $u_0 \approx_E u_1 \approx_E \cdots \approx_E u_n$ where $n \geq 0$, $u = u_0$, $v = u_n$, and $u_{i-1} \approx_E u_i$ for all $i > 0$. Given a Thue system $E$ and a pair of words $u$ and $v$, the *uniform word problem for Thue Systems* is the problem of deciding whether $u \approx_E v$. This is also called the *word problem for semigroups*, although in this case the problem is stated semantically [3]. The syntactic version of the word problem for semigroups was shown equivalent to the semantic version by Birkhoff.

Another way to examine the problem is to view the members of $A$ as monadic function symbols. In that case, a set of variables $V$ is added to the language. We refer to members of $V$ with letters $x, y, z$. Also, a set of constants $C$ is added. A *term* is a variable, or a constant, or a function symbol applied to a term (parentheses are omitted). Equations are of the form $s \approx t$, where $s$ and $t$ are terms. A *substitution* is a mapping from the set of variables to the set of terms, which is the identity almost everywhere. A substitution is extended to its homomorphic extension (i.e., $(ft)\sigma = f(t\sigma)$). A *ground instance* of a term (resp. equation) $t$ is anything of the form $t\sigma$, where $x\sigma$ is ground for all $x$ in $t$. If $t$ is a term or equation, then $Gr(t)$ is the set of all ground instances of $t$. If $E$ is a set of equations, then $v \approx_E v$ if and only if every ground instance of $u \approx v$ is in the congruence closure of the set of all ground instances of equations in $E$. That is the semantic definition. This also could be defined syntactically by saying that $us \approx_E ut$ if $s \approx t$ is an instance of an equation of $E$. Proofs are defined as for Thue Systems, and the word problem is stated in the same way. Birkhoff showed that the semantic and syntactic definition are equivalent. In this paper the syntactic definition will be more useful.

Given terms $u$ and $v$ and a set of equations $E$, $\sigma$ is an *$E$-unifier* of $u$ and $v$ if $u\sigma \approx_E v\sigma$. The unification problem for monadic function symbols is to find all $E$-unifiers of $u$ and $v$. A set of equations $E$ is said to be *unitary* if for every pair of terms $u$ and $v$, there is one $E$-unifier $\sigma$ such that for every $E$-unifier $\theta$ there

[1] See [1] for the case where $u = \epsilon$ or $v = \epsilon$.

[2] Therefore one relation Thue Systems are presented with two equations.

[3] $u \approx_E v$ if and only if $u \approx v$ is true in every model of $E$

is a substitution $\eta$ such that $x\sigma\eta \approx_E x\theta$ for every variable $x$ in $u$ or $v$. Then $\sigma$ is a *most general unifier* of $u$ and $v$.

It is well-known that the word problem for Thue Systems can be expressed as a word problem for monadic function symbols. Given the word problem for Thue Systems $u \approx_E v$, where $E = \{s_1 \approx t_1, \cdots, s_n \approx t_n\}$, we transform it into a word problem for monadic function symbols by asking if $uc \approx_E vc$, such that $E = \{s_1 x \approx t_1 x, \cdots, s_n x \approx t_n x\}$, where $x$ is a variable and $c$ is a constant.[4]

We need the notion of rewriting in terms of words. Let $s$ and $t$ be words (possibly empty), then $s \rightarrow t$ is a *rewrite rule.* If $R$ is a set of rewrite rules, then we write $usv \rightarrow utv$, and say *$usv$ rewrites to $utv$* if $s \rightarrow t \in R$. The reflexive transitive closure of $\rightarrow$ is written as $\rightarrow^*$. A word $u$ is in *$R$- normal form* if there is no $v$ such that $u$ rewrites to $v$. A set of rewrite rules is *confluent* if $s \rightarrow^* t$ and $s \rightarrow^* u$ implies that there is a $v$ such that $t \rightarrow^* v$ and $u \rightarrow^* v$. A set of rewrite rules $R$ is *weakly terminating* if for every $u$ there is a $v$ in normal form such that $u \rightarrow^* v$. $R$ is *strongly terminating* if there is no infinite rewrite sequence. A confluent and strongly terminating set of rewrite rules has the property that every rewrite sequence from $u$ leads to the same $v$ in normal form. We say that a set of rewrite rules $R$ is *non-overlapping* if there are no rules $s \rightarrow t$ and $u \rightarrow v$ such that a nonempty prefix of $s$ is a suffix of $u$ or $s$ is a subword of $u$. If a set of rewrite rules $R$ is non-overlapping and weakly terminating then $R$ is confluent and strongly terminating.

## 3 The Word and Unification Problem

In this section we give a class of Thue systems which is a generalization of one equation Thue systems. Then we give the structure of a proof of a unification (or word) problem in this generalized class. Finally, we show how this proof structure leads us to a problem in rewrite systems.

First we give the generalized class. A key idea is the notion of syntacticness from [5].

**Definition 1.** *A proof $u_0 \approx u_1 \approx \cdots \approx u_n$ is* syntactic *if there is at most one $i$ such that $u_{i-1} \approx_E u_i$ is an equational step at the top. A Thue System $E$ is* syntactic *if whenever there is a proof of $u \approx_E v$, then there is a syntactic proof of $u \approx_E v$.*

There is another restriction we need on the class to allow for our final procedure to be deterministic.

**Definition 2.** *A Thue System $s_1 \approx t_1, \cdots, s_n \approx t_n$ has* a repeated top equation *if there is an $i \neq j$ and $a, b \in A$ and words $u, v, u', v'$ such that $s_i = au$, $t_i = bv$ and $s_j = au'$, $t_j = bv'$. A Thue System $s_1 \approx t_1, \cdots, s_n \approx t_n$ has* a repeated top symbol *if there is an $i$ and $j$ $(i \neq j)$ and $a \in A$ and words $u, v$ such that $s_i = au$ and $s_j = av$, or if there is an $i$ and an $a$ and an $s_i$ and $t_i$ and words $u$ and $v$ such that $s_i = au$ and $s_j = av$.*

[4] We sometimes confuse the notation of words and terms. When the distinction is important, we clarify it.

Every word problem for Thue systems of one relation can be reduced to a simpler word problem which is either known to be solvable or has a different top symbol on the left and right side [3]. Therefore, for one equation Thue systems, it is only necessary to consider word problems where the one equation is of the form $as \approx bt$, where $a$ and $b$ are different symbols and $s$ and $t$ are words. Such theories are syntactic [2]. Such theories also have no repeating equation. Below, we show that any theory with no repeating top symbol is syntactic and has no repeating top equation.

**Theorem 1.** *Let $E$ be a Thue system that has no repeating top symbol. Then $E$ has no repeating top equation and $E$ is syntactic*

*Proof.* The fact that $E$ has no repeating top equation follows by definition. We prove that $E$ is syntactic by contradiction. Consider the set of all shortest proofs between any pair of terms. Consider the shortest proof $u_0 \approx_E u_1 \cdots \approx_E u_n$ of that set with more than one equational step at the top. Then there is a step from some $u_i$ to $u_{i+1}$ at the top using some equation $au \approx bv$, and later there is another step from some $u_j$ to $u_{j+1}$ using $bv \approx au$. Since this is the shortest proof, every proper subproof must be syntactic. But then there can be no intermediate steps involving $u$ and $v$, so the steps from $u_i$ to $u_{i+1}$ and from $u_j$ to $u_{j+1}$ can be removed from the proof resulting in a shorter proof of $u_1 \approx_E u_n$.

The results in this paper apply to syntactic Thue systems with no repeating top equation. Next we look at the structure of proofs of the unification problem in such theories. First we consider the case of unifying two terms with a different top symbol.

**Lemma 1.** *Let $E$ be a syntactic Thue System. Let $aux$ and $bvy$ be terms. If $\sigma$ is an $E$-unifier of $aux$ and $bvy$ then $\sigma$ is of the form $[x \mapsto u'z, y \mapsto v'z]$ and there exists*

- *an equation $asx \approx btx \in E$,*
- *words $r_1, r_2$ such that $u' = r_1r_2$,*
- *words $w_1, w_2$ such that $v' = w_1w_2$,*
- *and words $s', t'$ such that $ur_1 \approx_E ss'$, $tt' \approx_E vw_1$, and $s'r_2 \approx_E t'w_2$.*

*Proof.* Let $u_0 \approx_E \cdots \approx_E u_n$ be the proof of $aux\sigma \approx_E bvy\sigma$. There must be exactly one equational step at the top. The proof can be divided up into four parts. First $aux\sigma$ must be changed to a new word with $as$ as prefix. The second part is to change $as$ to $bt$. The third part is all the steps below $bt$, and the fourth part is to change $bt$ into a word with $bv$ as prefix. The second and third parts can be exchanged, but wlog we assume they happen in the order given.

Suppose that $u_i \approx_E u_{i+1}$ is the first equational step at the top. Then $u_i$ has $as$ as a prefix. This means that $x\sigma$ must have some $r_1$ as a prefix, such that there is an $s'$ where $ur_1 \approx_E ss'$. Therefore $u_{i+1}$ has $bts'$ as a prefix. That gives us the first part of the proof. The third part is all the steps below $bt$ so there must be $r_2, t', w_2$ such that $s'r_2 \approx_E t'w_2$. The fourth part changes $tt'$ to something with a $v$ as a prefix, so there must be a $w_1$ such that $tt' \approx_E vw_1$.

To sum it all up, the proof looks like: $aur_1r_2 \approx_E ass'r_2 \approx_E bts'r_2 \approx_E btt'w_2 \approx_E bvw_1w_2$.

Now we look at the proof structure when unifying two terms with the same top symbol.

**Lemma 2.** *Let $E$ be a syntactic set of monadic equations containing no equation of the form $as = at$. Then $\sigma$ is an $E$-unifier of $u$ and $v$ if and only if $\sigma$ is an $E$-unifier of $au$ and $av$.*

*Proof.* Suppose there is an equational step at the top of the proof of $au \approx_E bv$. Since there is no equation of the form $as \approx at$, there must be two equational steps at the top of the proof. But that cannot be, because $E$ is syntactic. Therefore, there is no equational step at the top of the proof.

Note that the condition of no equation of the form $as \approx at$ is implied by the condition of no repeated equation, since each Thue System is assumed to be closed under symmetry.

Our next step is to convert each unification problem to a rewrite system over an extended language, where the above two lemmas are applied as rewrite rules. First we define a new alphabet $\bar{A} = \{\bar{a} \mid a \in A\}$. Let $B = A \cup \bar{A}$. We define an inverse function on words in $B^*$ such that

- If $a \in A$ then $a^{-1} = \bar{a}$.
- If $\bar{a} \in \bar{A}$ then $(\bar{a})^{-1} = a$.
- $(b_1 \cdots b_n)^{-1} = {b_n}^{-1} \cdots {b_1}^{-1}$ for $n \geq 0$, and $b_i \in B$ for all $1 \leq i \leq n$.

Any word $w \in B^*$ can be represented uniquely in the form $u_1{v_1}^{-1} \cdots u_n{v_n}^{-1}$ where $n \geq 0$, each $u_i, v_i \in A^*$, $u_i = \epsilon$ only if $i = 1$ and $v_i \neq \epsilon$ , and similarly $v_i = \epsilon$ only if $i = n$ and $u_i \neq \epsilon$. We say that $w$ has $n$ *blocks.*

Given a Thue System $E = \{a_1s_1 \approx b_1t_1, \cdots, a_ns_n \approx b_nt_n\}$ over $A$, we define a rewrite system $R_E$ over $B$ containing

- $\bar{a}_ib_i \rightarrow s_i{t_i}^{-1}$ for all $1 \leq i \leq n$, and
- $\bar{a}a \rightarrow \epsilon$ for all $a \in A$. These are called *cancellation rules.*

*Example 1.* Given the Thue System $\{abaa \approx bbba, bbba \approx abaa\}$, the associated rewrite system $R_E$ is

1. $\bar{a}b \rightarrow baa\bar{a}\bar{b}\bar{b}$
2. $\bar{b}a \rightarrow bba\bar{a}\bar{a}\bar{b}$
3. $\bar{a}a \rightarrow \epsilon$
4. $\bar{b}b \rightarrow \epsilon$

Given an $E$-unification problem $G = ux \approx vy$, we associate a word $w_G = u^{-1}v$. Then we have the following theorem which translates a unification problem into a rewrite problem.

**Theorem 2.** *Let $E$ be a syntactic Thue System with no repeating equation, and $G$ be a unification problem over $A$. Then $G$ has a solution if and only if $w_G$ has an $R_E$-normal form of the form $u'(v')^{-1}$ with $u'$ and $v'$ in $A^*$.*[5] *Furthermore $\sigma = [x \mapsto u'z, y \mapsto v'z]$ is the most general unifier of $G$ in $E$.*

*Proof.* Given a word $w$ of $n$ blocks, where $w = u_1{v_1}^{-1} \cdots u_n{v_n}^{-1}$, we think of $w$ as representing the unification problem $(x = u_1z_1) \wedge \bigwedge_{2\leq i\leq n}(v_{i-1}z_{i-1} = u_iz_i) \wedge (v_nz_n = y)$ for some $z_1, \cdots, z_n$. Therefore, if $G = ux \approx_E vy$, then $w_G = u^{-1}v$ represents the unification problem $x = z_1 \wedge uz_1 = vz_2 \wedge z_2 = y$, that is $ux \approx_E vy$.

Since each word $w$ represents a unification problem, the solution to the unification problem has a corresponding proof. We will give an induction argument based on the lexicographic combination of the length of that corresponding proof and the number of symbols in $w$.

Suppose we are given a word $w = u^{-1}v$, representing the unification problem $ux \approx_E vy$. If $ux = au_1x$ and $vy = av_1y$ for some $u_1$ and $v_1$, then $ux \approx_E vy$ has most general unifier $\sigma$ if and only if $u_1x \approx_E v_1y$ has most general unifier $\sigma$. Then the word $w = u^{-1}v = {u_1}^{-1}\bar{a}av_1 \to {u_1}^{-1}v_1$, which is a smaller unification problem.

Suppose that $ux = au_1x$ and $vy = bv_1y$, and suppose there is an equation $as \approx bt \in E$. Then $\bar{a}b \to st^{-1}$. Furthermore, the unification problem $ux \approx_E vy$ is satisfiable and has most general unifier $\sigma = [x \mapsto u'z, y \mapsto v'z]$ if and only if there are words $r_1, r_2$ such that $u' = r_1r_2$, words $w_1, w_2$ such that $v' = w_1w_2$, and words $s', t'$ such that $u_1r_1 \approx_E ss'$, $tt' \approx_E v_1w_1$, and $s'r_2 \approx_E t'w_2$. So ${u_1}^{-1}s \to r_1{s'}^{-1}$, $t^{-1}v_1 \to t'{w_1}^{-1}$, and ${s'}^{-1}t' \to r_2{w_2}^{-1}$, Then the word $w = u^{-1}v = {u_1}^{-1}\bar{a}bv_1 \to {u_1}^{-1}st^{-1}v_1 \to r_1{s'}^{-1}t'{w_1}^{-1} \to r_1r_2{w_2}^{-1}{w_1}^{-1}$. This is in normal form, and it represents the unification problem $x = r_1r_2z_1 \wedge w_1w_2z_1 = y$, which has $\sigma$ as most general unifier.

Suppose that $ux = au_1x$ and $vy = bv_1y$, and there is no equation $as \approx bt \in E$. Then $w = u^{-1}v = {u_1}^{-1}\bar{a}bv_1$ has no redex at the subword $\bar{a}b$, and therefore has no normal form with zero or one block.

This theorem also shows that any syntactic Thue system with no repeating equation has a unitary $E$-unification problem, because the rewriting is deterministic and leads to at most one most general unifier.

The following corollary shows how the theorem applies to word problems.

**Corollary 1.** *Let $E$ be a syntactic Thue System with no repeated equation, and $G$ be a word problem over $A$. Then $G$ is true in $E$ if and only if the normal form of $w_G$ in $R_E$ is $\epsilon$.*

*Proof.* The corollary follows from the theorem because the word problem is true if and only if $\sigma = [x \mapsto z, y \mapsto z]$ is a most general unifier.

*Example 2.* For example, consider the Thue System $\{aa \approx ba, ba \approx aa\}$. Then $R_E$ is

[5] Notice that if $A$ has two symbols, then every normal form is of this form.

1. $\bar{a}b \rightarrow a\bar{a}$
2. $\bar{b}a \rightarrow a\bar{a}$
3. $\bar{a}a \rightarrow \epsilon$
4. $\bar{b}b \rightarrow \epsilon$

Let $G$ be the unification problem $abb \approx_E bb$. Then $w_G = \bar{b}\bar{b}\bar{a}bb$. This gives us the following rewrite sequence: $w_G = \bar{b}\bar{b}\bar{a}bb \rightarrow \bar{b}\bar{b}a\bar{a}b \rightarrow \bar{b}\bar{b}aa\bar{a} \rightarrow \bar{b}a\bar{a}a\bar{a} \rightarrow \bar{b}a\bar{a} \rightarrow a\bar{a}\bar{a}$, which is in normal form. That means that the most general unifier of $G$ in $E$ is $\sigma = [x \mapsto az, y \mapsto aaz]$. The word problem for $G$ is not true, because the normal form is not $\epsilon$. However, if we consider the word problem $G' = abba \approx bbaa$, then $w_{G'} = \bar{a}w_G aa \rightarrow^* \bar{a}a\bar{a}\bar{a}aa \rightarrow^* \epsilon$. Therefore, the word problem $G'$ is true in $E$.

*Example 3.* For another example, consider the Thue System $E = \{a \approx bba, bba \approx a\}$. Then $R_E$ is

1. $\bar{a}b \rightarrow \bar{a}\bar{b}$
2. $\bar{b}a \rightarrow ba$
3. $\bar{a}a \rightarrow \epsilon$
4. $\bar{b}b \rightarrow \epsilon$

Consider the unification problem $G = ba \approx_E a$. Then $w_G = \bar{a}\bar{b}a \rightarrow \bar{a}ba \rightarrow \bar{a}\bar{b}a = w_G$. So this rewrite sequence loops, and there are no other possible rewrite sequences. Therefore $\bar{a}\bar{b}a$ has no normal form in $R_E$ which implies that the unification problem $ba \approx_E a$ (and also the word problem) has no solution in $E$.

## 4 Deciding the Unification Problem

In this section we first show why the condition of syntacticness and no repeating equations leads to a deterministic procedure. Then we define an ordering to show termination of rewrite sequences. This ordering is used to define a decision procedure for certain classes of problems. In some cases, we can even bound the complexity by an exponential of the goal size. Finally, we give an algorithm that we conjecture decides the unification problem in all cases.

First we show the interest of the class of problems we are considering.

**Theorem 3.** *If $E$ is a syntactic Thue System with no repeating top equation, and $G$ is a satisfiable unification problem in $E$, then $R_E$ is confluent and strongly terminating on $w_G$.*

*Proof.* It follows from the fact that $R_E$ is non-overlapping and is weakly terminating on $w_G$.

This gives us a deterministic procedure to decide the word problem. We can assume that we always apply the rightmost rewrite step. Unfortunately, this deterministic procedure may not always halt. So we need some ways to determine non-termination or rewrite sequences. One way to do this is to find loops. First, let's define a useful ordering to determine terminating rewrite sequences.

We define an ordering on words of three or fewer blocks.

**Definition 3.** *Let $w$ be an $n$ block word of the form $u_1{v_1}^{-1} \cdots u_n{v_n}^{-1}$, with each $u_i, v_i \in A^*$ and $n \leq 3$. Define $\mu(w)$ to be the ordered pair $(i, j)$ such that if $n \geq 1$ then $i = |v_1|$ else $i = 0$, and if $n = 3$ then $j = |u_3|$ else $j = 0$. Ordered pairs are compared lexicographically, i.e., $(i, j) > (k.l)$ if and only if $i > k$, or $i = k$ and $j > l$. Note that this ordering is well-founded.*

Now we define a set of words that we will later use to show that if we can find the normal forms of these words, then we can find the normal form of any given word, or determine that it does not have one.

**Definition 4.** *Let $A$ be a set of words and $R$ be a rewrite system on $B = A \cup \bar{A}$. Let $C$ be a set of words in $(\bar{A})^*$, such that every non-empty prefix of $C$ is in $C$. A word $ua$ is called an* extended word of $C$ *if $u \in C$ and $a \in A$. Let $C'$ be the set of all $R$-normal forms $w$ of extended words of $C$. Then $R$ is $C$-complete if for all $w$ in $C$*

1. *$w$ contains one at most one block, and*
2. *if $w$ contains one block (i.e., $w = u_1 v_1$ with $u_1 \in A^*$ and $v_1 \in (\bar{A})^*$) then $v_1$ is in $C$ if $v_1 \neq \epsilon$.*

Note that condition 2 is trivially true if $a$ is the inverse of the last letter in $u$.

**Definition 5.** *Let $R$ be a rewrite system of the form $\{\bar{a_1} b_1 \to s_1{t_1}^{-1} \cdots \bar{a_n} b_n \to s_n{t_n}^{-1}\}$, with each $a_i, b_i \in A$, and $s_i, t_i \in A^*$. $C$ is said to be a* completion of $R$ *if $R$ is $C$-complete and ${t_i}^{-1} \in C$ if $t_i \neq \epsilon$.*

If we rewrite in a certain way, we can force one of these special words to appear in a certain place in the word.

**Definition 6.** *Let $C$ be a set of words. The word $w$ is $C$-*reducible *if and only if $w$ has at most three blocks, and if $w$ has three blocks (i.e., is of the form $u_1{v_1}^{-1}u_2{v_2}^{-1}u_3{v_3}^{-1}$) then ${v_2}^{-1} \in C$.*

We use the previous four definitions in the following crucial lemma. It shows that any word of a particular form can be reduced to a smaller word of the same form, or we can detect that it will not have an appropriate normal form.

**Lemma 3.** *Let $E$ be a Thue System and $C$ be a completion of $R_E$. Let $w$ be a $C$-reducible word in $B^*$ of three or fewer blocks. Suppose that for all extended words $ua$ of $C$, it is decidable whether $ua$ has an $R_E$-normal form. If $w$ is not in normal form, then we can find a smaller $C$-reducible $w'$ with three or fewer blocks such that $w \to w'$ or we can detect that $w$ has no normal form with one or fewer blocks.*

*Proof.* If $w$ has at most one block, then $w$ is in normal form. Suppose $w$ has two blocks, then $w = u_1{v_1}^{-1}u_2{v_2}^{-1}$ with $u_1, v_1, u_2, v_2 \in A^*$. Suppose $v_1 = av$ and $u_2 = bu$. Then $w = u_1 v^{-1} \bar{a} b u {v_2}^{-1}$. If $a = b$ then $w \to u_1 v^{-1} u {v_2}^{-1} = w'$, which is smaller than $w$ because $\mu(w) = (|v_1|, 0) > (|v|, 0) = \mu(w')$. Also, $w'$ has fewer than three blocks, so it is $C$-reducible.

Suppose $a \neq b$ and there is a rule in $R_E$ of the form $\bar{a}b \rightarrow st^{-1}$. Then $w \rightarrow u_1 v^{-1} st^{-1} u {v_2}^{-1} = w'$, which is smaller than $w$ because $\mu(w) = (|v_1|, 0) > (|v|, |u|) = \mu(w')$. [6] $w'$ has three or fewer blocks. Also, note that $t^{-1} \in C$ by definition of $C$-complete, therefore $w'$ is $C$-reducible.

If $a \neq b$ and there is no rule in $R_E$ of the form $\bar{a}b \rightarrow st^{-1}$, then any normal form of $w$ must have more than one block.

Now suppose $w$ has three blocks, then $w = u_1 {v_1}^{-1} u_2 {v_2}^{-1} u_3 {v_3}^{-1}$ with $u_1, v_1, u_2, v_2, u_3, v_3 \in A^*$. Suppose $v_2 = av$ and $u_3 = bu$. Then $w = u_1 {v_1}^{-1} u_2 v^{-1} \bar{a} b u {v_3}^{-1}$. If $a = b$ then $w \rightarrow u_1 {v_1}^{-1} u_2 v^{-1} u {v_3}^{-1} = w'$, which is smaller than $w$ because $\mu(w) = (|v_1|, |u_3|) > (|v_1|, |u|) = \mu(w')$. $w'$ has at most three blocks. ${v_2}^{-1} \in C$ since $w$ is $C$-reducible, therefore $v^{-1} \in C$ since $C$ is closed under prefixes, so $w'$ is $C$-reducible.

Suppose $a \neq b$. Then, ${v_2}^{-1} \in C$, because $w$ is $C$-reducible. So we can decide whether $v_2 b$ has an $R_E$-normal. form. If it has no $R_E$-normal form then neither does $w$. Otherwise, we can calculate the normal form of $v_2 b$. By definition of $C$-complete, the normal form $v_2 b$ is of the form $u' {v'}^{-1}$, with ${v'}^{-1} \in C$.[7] Then $w \rightarrow u_1 {v_1}^{-1} u_2 u' {v'}^{-1} u {v_3}^{-1} = w'$, which is smaller than $w$ because $\mu(w) = (|v_1|, |u_3|) > (|v_1|, |u|) = \mu(w')$. $w'$ has at most three blocks. Also, $w'$ is $C$-reducible, since ${v'}^{-1} \in C$.

If $a \neq b$ and there is no rule in $R_E$ of the form $\bar{a}b \rightarrow st^{-1}$, then any normal form of $w$ must have more than one block.

The following theorem is the main result used to decide the word and unification problem.

**Theorem 4.** *Let $E$ be a Thue System and $G$ a goal over $A$. Let $C$ be a completion of $R_E$.*

1. *Suppose that for all extended words $ua$ of $C$ it is decidable whether $ua$ has an $R_E$-normal form. Then the word and unification problem for $E$ is decidable.*

2. *If $C$ is finite, then the word and unification problem is decidable in time at most exponential in the size of the goal.*

*Proof.* We construct $R_E$ and $w_G$. Note that $w_G$ has two blocks. Let $w$ be a $C$-reducible word with three or fewer blocks. We perform induction on $\mu(w)$. The induction hypothesis is that we can find the normal form of all smaller words or prove they do not have one with one or fewer blocks. By the previous lemma, we can either reduce $w$ to a smaller $C$-reducible $w'$ with three or fewer blocks, or else detect that $w$ has no normal form with one or fewer blocks. In the second

[6] In all of these cases, we should consider the case where $u = \epsilon$, but then $\mu(w') < \mu(w)$ because the second number in the ordered pair of $\mu(w')$ is 0.

[7] Here we do not consider the simpler cases where the normal form is $\epsilon$ or only contains members of $A$ or $A^{-1}$.

case, we are done. In the first case, $w$ has the same normal form as $w'$, so we are also done.

This takes care of the first part of the theorem. When $C$ is finite, the above argument still shows that the word problem is decidable, since decision problems on finite sets are always decidable. But we must show that the decision procedure runs in at most exponential time in the size of the goal. For that we must analyze the procedure induced by the previous lemma. If $\mu(w) = (i, j)$, then there are at most $j$ rewrite steps before $i$ gets smaller. But during that time, $w$ can increase by a product of $k$, where $k$ is the maximum size of $u_1$ for a normal form $u_1 {v_1}^{-1}$ of $ua$ with $u \in C$. Therefore, to calculate the normal form of $w$, we potentially multiply $w$ by $k$, $|w|$ times, at most. So the word can become as big as $k^{|w|}$ at most. And since each operation is linear in the size of the goal, the running time as also bounded by an exponential.

We give some examples to illustrate.

*Example 4.* Let $E = \{aba \approx bab, bab \approx aba\}$. Then $R_E$ is

1. $\bar{a}b \to ba\bar{b}\bar{a}$
2. $\bar{b}a \to ab\bar{a}\bar{b}$
3. $\bar{a}a \to \epsilon$
4. $\bar{b}b \to \epsilon$

Let $C = \{\bar{b}, \bar{a}, \bar{b}\bar{a}, \bar{a}\bar{b}\}$. The normal forms of $\bar{b}a$, $\bar{a}b$, $\bar{b}\bar{a}b$ and $\bar{a}\bar{b}a$ are respectively $ab\bar{a}\bar{b}$, $ba\bar{b}\bar{a}$, $a\bar{b}\bar{a}$, and $b\bar{a}\bar{b}$. Each of these normal forms contains only one block. Since all nonempty prefixes of $\bar{a}\bar{b}$ and $\bar{b}\bar{a}$ are in $C$, then $C$ is a completion of $R_E$.

*Example 5.* Let $E = \{abb \approx baa, baa \approx abb\}$. Then $R_E$ is

1. $\bar{a}b \to bb\bar{a}\bar{a}$
2. $\bar{b}a \to aa\bar{b}\bar{b}$
3. $\bar{a}a \to \epsilon$
4. $\bar{b}b \to \epsilon$

Let $C = \{\bar{a}, \bar{b}, \bar{a}\bar{a}, \bar{b}\bar{b}\}$. The normal forms of $\bar{a}b$ and $\bar{b}a$ are respectively $bb\bar{a}\bar{a}$ and $aa\bar{b}\bar{b}$. Note that $\bar{a}\bar{a}b \to \bar{a}bb\bar{a}\bar{a} \to bb\bar{a}\bar{a}b\bar{a}\bar{a}$ which contains $\bar{a}\bar{a}b$ as subword. Therefore $\bar{a}\bar{a}b$ has no normal form. Similarly, $\bar{b}\bar{b}a$ has no normal form. We only need to consider the normal forms $bb\bar{a}\bar{a}$ and $aa\bar{b}\bar{b}$. Since all the nonempty prefixes of $\bar{a}\bar{a}$ and $\bar{b}\bar{b}$ are in $C$, then $C$ is a completion of $R_E$.

Here is an example for which $C$ is infinite.

*Example 6.* Let $E = \{bab \approx a, a \approx bab\}$. Then $R_E$ is

1. $\bar{b}a \to ab$
2. $\bar{a}b \to \bar{b}\bar{a}$
3. $\bar{a}a \to \epsilon$
4. $\bar{b}b \to \epsilon$

Let $C = \{(\bar{b})^n \mid n > 0\} \cup \{(\bar{b})^n\bar{a} \mid n \geq 0\}$. Given $n$, the normal form of $(\bar{b})^n a$ is $ab^n$, and the normal form of $(\bar{b})^n\bar{a}b$ is $(\bar{b})^{n+1}\bar{a}$. These can be proved by induction on $n$. The cases where $n = 0$ are trivial. If $n > 0$, we have $(\bar{b})^n\bar{a}b = \bar{b}(\bar{b})^{n-1}\bar{a}b \rightarrow (\bar{b})^n\bar{a}$. Also, $(\bar{b})^n a = \bar{b}(\bar{b})^{n-1}a \rightarrow \bar{b}ab^{n-1} = ((\bar{b})^{n-1}\bar{a}b)^{-1} \rightarrow ((\bar{b})^n\bar{a})^{-1} = ab^n$.

Here we used the fact that $u^{-1} \rightarrow v^{-1}$ if $u \rightarrow v$.

Now we must address the question of how to determine if a word has a nonterminating rewrite sequence. We say a word $w$ *loops* if there exist words $u$ and $v$ such that $w \rightarrow^+ uwv$. We conjecture that every nonterminating rewrite sequence loops. This is the same as the conjecture for one rule semi-Thue systems in [10].

*Conjecture 1.* Let $E$ be a syntactic Thue System with no repeated equations, and $G$ be a unification problem. Then $w_G$ has a nonterminating rewrite sequence in $R_E$ if and only if there are some words $u, v, w$ such that $w_G \rightarrow uvw$ and $v$ loops.

It is possible to detect loops, so a proof of the conjecture would imply that the unification and word problem are decidable. We now give an algorithm for deciding the unification problem, whose halting relies on the truth of the conjecture.

For the algorithm, we are given a Thue System $E$, and a goal $G$. We construct $R_E$ and $w_G$. The intention of the algorithm is to reduce the goal to its normal form at the same time we are creating a subset of the extensions of $C$ (the completion of $R_E$), and keeping track of the normal forms or lack of normal forms of those extensions of $C$.

The algorithm involves $w$ which is initially set to $w_G$ and any applicable cancellation rules are applied. $w$ is always a reduced version of $w_G$ with at most three blocks. The algorithm also involves a stack $T$ of ordered pairs. Each element of $T$ is an ordered pair $(u, v)$ such that $u$ is of the form ${u'}^{-1}a$ with $u' \in A^*$ and $a \in A$, and $v$ is a word of at most three blocks. The values of $u$ will be words that we are trying to find the normal form of, and $v$ will be a reduced version of $u$. There is a set of ordered pairs $S$ involved in the algorithm. An element of $S$ is an ordered pair $(u, v)$ where $u$ is of the form ${u'}^{-1}a$ with $u' \in A^*$ and $a \in A$, and $v$ is a word of one or fewer blocks which is the normal form of $u$. $S$ and $T$ are both initially empty.

The algorithm proceeds as follows:

First check if $T$ is empty. If $T$ is empty and $w$ is in normal form, then check if $w$ has one or fewer blocks. If it does, then return $w$. That is the normal form of $w_G$. If it does not, then return FALSE, because $w_G$ has no normal form of one or fewer blocks, thus the unification problem is false.

Suppose $T$ is empty and $w$ is not in normal form, we examine the rightmost redex position of $w$. Either $w$ has two blocks and is of the form $u_1{v_1}^{-1}u_2{v_2}^{-1}$, or $w$ has three blocks and is of the form $u_1{v_1}^{-1}u_2{v_2}^{-1}u_3{v_3}^{-1}$. If $w$ has two blocks, set $u' = {v_1}^{-1}$ and set $c$ to be the first letter of $u_2$. If $w$ has three blocks, set $u' = {v_2}^{-1}$ and set $c$ to be the first letter of $u_3$. If $\bar{d}$ is the last character in $u'$ and there is no $c$ such that $\bar{d}c \in R_E$, then return FALSE. Search for an ordered pair

$(u'c, v)$ in $S$ for some $v$. If it exists, then replace $u'c$ in $w$ by $v$ and perform any cancellation rules that now apply. Note that $w$ still has at most three blocks. If no $(u'c, v)$ exists in $S$, then push $(u'c, u'c)$ onto $T$.

If $T$ is not empty, let $(u, v)$ be on top of the stack. Either $v$ has two blocks and is of the form $u_1{v_1}^{-1}u_2{v_2}^{-1}$, or $v$ has three blocks and is of the form $u_1{v_1}^{-1}u_2{v_2}^{-1}u_3{v_3}^{-1}$. If $v$ has two blocks, set $u' = {v_1}^{-1}$ and set $c$ to be the first letter of $u_2$. If $v$ has three blocks, set $u' = {v_2}^{-1}$ and set $c$ to be the first letter of $u_3$. If $\bar{d}$ is the last character in $u'$ and there is no $c$ such that $\bar{d}c \in R_E$, then return FALSE. Search for an ordered pair $(u'c, v')$ in $S$ for some $v'$. If it exists then replace $u'c$ in $v$ by $v'$ and perform applicable cancellations. Note that $v$ still has at most three blocks. If $v$ is now in normal form, then if $v$ has at most one block, then we add $(u, v)$ to $S$ and remove $(u, v)$ from $T$, else return FALSE. If $v$ contains $u$ as a subword, or if $v$ contains $s$ as a subword with $(s, t)$ in $T$ for some $T$, we return FALSE. If no $(u'c, v')$ exists in $S$, then push $(u'c, v)$ onto $T$.

Keep repeating this process until it halts.

Based on our implementation, this algorithm appears to be very efficient, and we conjecture that it always halts. Note that the algorithm constructs extensions of a completion of $R_E$. Based on theorem 4, we can see that this algorithm will halt in time at most exponential in the size of the goal if $R_E$ has a finite completion.

There is another interesting generalization of the class of problems. We consider Thue systems to only contain equations of the form $ux \approx vx$. Suppose we allowed other monadic terms. For example $ux \approx vy$. If all our equations are of this type, then lemma 1 is still true, with the removal of the condition that $s'r_2 \approx_E t'w_2$. We could say a simliar thing for equations of the form $ua \approx vb$. This would allow us to modify the definition of $R_E$ so that the right hand side of the rewrite rules have a marker between the two halves, preventing interaction between the two. This allows us to solve the unification problem in polynomial time in terms of the goal if $E$ is a syntactic set of monadic terms, with no repeated equations, and no equations of the form $ux \approx vx$. Space prevents us from giving the details of this argument. But it is interesting to note that the problem becomes easier when the equations are not linear.

## 5 Conclusion

We have given a method for trying to solve the unification (and word) problem for one equation Thue systems and other monadic equational theories. Our method works on a larger class of problems, which we have defined. We have shown certain cases where we can prove that the method is a decision procedure. We gave an algorithm, which has been implemented, and appears to be efficient. It halts and serves as a decision procedure for every input we have tried. This is opposed to the Knuth-Bendix procedure which often runs forever. The closest work to our approach is given in [4]. This is based on an algorithm in [2] for Thue systems with one equation. The algorithm does not always halt. In [4], a rewrite system is given to help determine when the algorithm of [2] halts. They also needed to prove the termination of the rewrite system. But their method and rewrite system is quite different from ours. For example, our rewrite system halts on different problems than theirs. They also gave an example of a rewrite system with a word that did not terminate but had no loop (called a *simple loop* in their paper). It would be interesting to do a more detailed comparison of our two methods. We think that methods used to decide termination of one rule semi-Thue systems might be helpful for us. Our ultimate goal is to extend our method to all unification problems over terms, and find a large class of problems for which our approach halts. This approach in this paper was designed with that intention.

## References


1. S. Adian. Definining relations and algorithmic problems for groups and semigroups. *Trudy Matem. in-ta im. Steklova AN SSSR*, 85, 1996 (Russian).
2. S. Adian. Transformations of words in a semigroup presented by a system of defining relations. *Algebra i logika,* 15(6),611-621, 1976 (Russian).
3. S. Adian, and G. Oganesian. On the word and divisibility problems in semigroups with a single defining relation. *Izv. An. SSSR Ser. Matem.,* 42(2),219-225, 1978 (Russian).
4. J. Bouwsma. Semigroups Presented by a Single Relation. PhD dissertation at Pennsylvania State University, 1993.
5. C. Kirchner. Computing unification algorithms. In *Proceedings of the First Symposium on Logic in Computer Science*, Boston, 200-216, 1990.
6. G. Lallement. The word problem for Thue rewriting systems. In *Spring School in Rewriting*, ed. H. Comon and J. P. Jouannaud, Lecture Notes in Computer Science, 1994.
7. C. Lynch. Goal Directed Completion using SOUR Graphs. In *Proceedings of the Eighth International Conference on Rewriting Techniques and Applications (RTA)*, Sitges, Spain, June 2–4, 1997.
8. W. Magnus. Das Identitätsproblem für Gruppen mit einer definierenden Relation. *Math Ann.,* 106,295-307, 1932.
9. Y. Matisaevich. Simple examples of unsolvable associative calculi. *Dokl. Akad. Nauk. SSSR,* 173,1264-1266, 1967.
10. R. McNaughton. Well-behaved derivations in one-rule Semi-Thue Systems. Tech. Rep 95-15, Dept. of Computer Science, Rensselaer Polytechnic Unstitute, Troy, NY, Nov. 1995.
11. W. Savitch. How to make arbitrary grammars look like context-free grammars. *SIAM Journal on Computing,* 2(3),174-182, September 1973.