Some Notes on Basic Syntactic Mutation*

Christopher Bouchard, Kimberly A. Gero, and Paliath Narendran

University at Albany—SUNY (USA)
{cbou,kgero001,dran}@cs.albany.edu

Abstract

Unification modulo convergent term rewrite systems is an important research area with many applications. In their seminal paper Lynch and Morawska gave three conditions on rewrite systems that guarantee that unifiability can be checked in polynomial time (**P**). We show that these conditions are *tight*, in the sense that relaxing any one of them will "upset the applecart," giving rise to unification problems that are not in **P** (unless **P** = **NP**), and in doing so address an open problem posed by Lynch and Morawska. We also investigate a related decision problem: we show the undecidability of subterm-collapse for the restricted term rewriting systems that we are considering.

Keywords: Equational unification, Term rewriting, Polynomial-time complexity, NP-completeness, Undecidability.

1 Introduction

Unification modulo an equational theory E (equational unification or E-unification) is an undecidable problem in general. Even in cases where it is decidable, it is often of high complexity. In their seminal paper "Basic Syntactic Mutation" [7] Christopher Lynch and Barbara Morawska present syntactic criteria on equational axioms E that guarantee a polynomial time algorithm for the corresponding E-unification problem. As far as we know these are the only purely syntactic criteria that ensure a polynomial-time algorithm for unifiability. Our goal initially was to extend the Lynch-Morawska result *for convergent term rewriting systems* by relaxing their constraints, while still maintaining the polynomial time algorithm guarantee. However, we observed that their constraints were tight in the sense that relaxing any one of them would give rise to unification problems that are not in **P** (unless **P** = **NP**). We provide proofs that removing any of their constraints will lead to unification problems that are not in **P**.

We also investigate one of the computational issues raised by the Lynch-Morawska paper, namely, checking whether a convergent term rewriting system is *subterm-collapsing* — a term rewriting system is subterm-collapsing if and only if there is a term that is congruent to a proper subterm of itself. We show the undecidability of subterm collapse even for convergent term rewriting systems that satisfy the Lynch-Morawska conditions [7]. (For general convergent systems, this result was shown by Bürkert, Herold and Schmidt-Schauß [4].)

2 Notation and Preliminaries

We assume the reader is familiar with the usual notions and concepts in term rewriting systems [2] and equational unification [3]. We consider rewrite systems over ranked signatures, usually denoted Σ, and a possibly infinite set of variables, usually denoted $\mathcal{X}$. The set of all

*Research supported in part by NSF grant CNS-0905286.

terms over Σ and $\mathcal{X}$ is denoted as $T(\Sigma, \mathcal{X})$. An *equation* is an ordered pair of terms (s, t), usually written as $s \approx t$. Here s is the left-hand side and t is the right-hand side of the equation [2]. A rewrite rule is an equation $s \approx t$ where $Var(t) \subseteq Var(s)$, usually written as $s \to t$. A term rewriting system is a set of rewrite rules.

A set of equations E is *subterm-collapsing*[1] if and only if there are terms t and u such that t is a proper subterm of u and $E \vdash t \approx u$ (or $t =_E u$) [4]. A set of equations E is *variable-preserving*[2] if and only if for every equation $t \approx u$ in E, $\mathcal{V}ar(t) = \mathcal{V}ar(u)$ [9]. A term rewriting system is *convergent* if and only if it is confluent and terminating [2].

Given a set of equations E, the set of ground instances of E is denoted by $Gr(E)$. We assume a reduction order $\prec$ on E which is total on ground terms. We extend this order to equations as $(s \approx t) \prec (u \approx v)$ iff $\{s, t\} \prec_{mul} \{u, v\}$, where $\prec_{mul}$ is the multiset order induced by $\prec$. An equation e is *redundant in* E if and only if every ground instance $\sigma(e)$ of e is a consequence of equations in $Gr(E)$ which are smaller than $\sigma(e)$ modulo $\prec$ [7].

2.1 Paramodulation

Lynch and Morawska define an extension to the critical pair rule called *paramodulation*. Since our focus is only on convergent term rewriting systems, this definition can be modified to use rewrite rules as

$$\frac{u[s']_p \approx v \qquad s \to t}{\sigma(u[t]_p) \approx \sigma(v)}$$

where $\sigma = mgu(s =^? s')$ and $p \in \mathcal{FP}os(u)$. A set of equations E is *saturated* if all inferences among equations in E are redundant.

2.2 Right-Hand-Side Critical Pairs

Lynch and Morawska also define a *right-hand-side critical pair*. Again, we modify the definition to use rewrite rules as

$$\frac{s \to t \qquad u \to v}{\sigma(s) \approx \sigma(u)}$$

where $\sigma = mgu(v =^? t)$ and $\sigma(s) \neq \sigma(u)$. For an equational theory E, we define $RHS(E)$ as follows [7]:

$$RHS(E) = E \cup \{\, e \mid e \text{ is the conclusion of a Right-Hand-Side Critical Pair inference of two members of } E \,\}$$

2.3 Quasi-determinism

A set of equations E is *quasi-deterministic* if and only if

(1) No equation in E has a variable as its left-hand side or right-hand side,

(2) No equation in E is *root-stable*—i.e., no equation has the same root symbol on its left- and right-hand side, and

(3) E has no *root pair repetitions*—i.e., no two equations in E have the same pair of root symbols on their sides.

[1] Non-subterm-collapsing theories are called *simple* theories in [4]

[2] Variable-preserving theories are also called *non-erasing* or *regular* theories [2].

A theory E is *deterministic* if and only if it is quasi-deterministic and non-subterm-collapsing.

Lemma 2.1. *Suppose R is a variable-preserving convergent rewrite system and R is quasi-deterministic. Then $RHS(R)$ is not quasi-deterministic if and only if $RHS(R)$ has a root pair repetition.*

Proof. If $RHS(R)$ has a root pair repetition, then clearly $RHS(R)$ is not quasi-deterministic by definition. We will prove the 'only if' case by contradiction.

Suppose $RHS(R)$ is not quasi-deterministic and has no root pair repetition. Since R itself is quasi-deterministic, there must be equations created in $RHS(R)$ that cause the non-quasi-determinism. Therefore there must be rules $l_1 \to r_1$ and $l_2 \to r_2$ in R such that $\theta(l_1) \approx \theta(l_2)$ is in $RHS(R)$. Since R is variable-preserving, it must be that $(l_1 \to r_1)$ and $(l_2 \to r_2)$ are distinct rules, for otherwise we would have $\theta(l_1) = \theta(l_2)$ and the generated rule would be discarded.

Since, by assumption, $RHS(R)$ has no root pair repetition, either one of l_1 or l_2 is a variable, or $\theta(l_1) \approx \theta(l_2)$ is root-stable. The first is not possible because rewrite rules cannot have a variable as their left-hand side. In the second case, rules $l_1 \to r_1$ and $l_2 \to r_2$ would cause a root pair repetition in R, which is a contradiction. □

2.4 Monotone 1-in-3-SAT

All of the NP-completeness proofs in this paper are shown by reductions from the *monotone 1-in-3-SAT problem* [2], which is known to be NP-complete [6]. The problem is defined as follows:

Instance: A CNF formula $\mathcal{F} = C_1 \wedge \cdots \wedge C_n$, where each clause is of the form $C_i = p_i \vee q_i \vee r_i$ and p_i, q_i, and r_i are propositional variables.

Question: Is there a satisfying assignment to the propositional variables of $\mathcal{F}$ that sets *exactly one* propositional variable of each clause to true and the other two to false?

3 Lynch-Morawska Conditions

Given a confluent and terminating term rewriting system R, there are three conditions that must hold to maintain a polynomial time algorithm guarantee:

(1) R is non-subterm-collapsing,
(2) R is saturated by paramodulation, and
(3) $RHS(R)$ is quasi-deterministic.

In this section we will show that if any one of these conditions is relaxed there is no longer a polynomial time guarantee. Therefore the conditions given in the Lynch-Morawska paper are tight.

3.1 R is Subterm-Collapsing

Consider the following single-rule ground term rewriting system R_1, where 0, 1, and c are constants.

$$f(0, f(0, f(1, c))) \to c$$

The system R_1 is saturated by paramodulation and $RHS(R_1)$ is quasi-deterministic. However, it is subterm-collapsing. The unification problem can be shown to be NP-hard by a reduction from monotone 1-in-3-SAT.

For each clause $C_i = p_i \vee q_i \vee r_i$ we form the following equation $\mathcal{EQ}_i$, where Z_i is a variable that varies with each clause.

$$f(V_{p_i}, f(V_{q_i}, f(V_{r_i}, Z_i))) =^?_{R_1} Z_i$$

The unification problem S is the set of all these equations. Every unifier of this equation replaces exactly one of V_{p_i}, V_{q_i}, or V_{r_i} by 1 and the others by 0. In fact,

$$\sigma_1 = \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 1,\, Z_i \mapsto c\}$$
$$\sigma_2 = \{V_{p_i} \mapsto 1,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 0,\, Z_i \mapsto f(1, c)\}$$
$$\sigma_3 = \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 1,\, V_{r_i} \mapsto 0,\, Z_i \mapsto f(0, f(1, c))\}$$

are the only normalized unifiers for the above equation. Clearly then S is unifiable iff the 1-in-3-SAT instance has a solution.

In [7], Lynch and Morawska posed, as an open problem, whether the requirement that R be non-subterm-collapsing can be removed; in other words, can the polynomial-time result be extended to systems which are saturated by paramodulation and quasi-deterministic but possibly subterm-collapsing? The rewrite system R_1 answers this problem negatively.

3.2 R is not Saturated by Paramodulation

Consider the following rewrite system R_2, where 0 and 1 are constants.

$$f_1(s(x)) \to f_2(x) \qquad f_1(0) \to g(0, 0, 1)$$
$$f_2(s(x)) \to f_3(x) \qquad f_2(0) \to g(0, 1, 0)$$
$$f_3(0) \to g(1, 0, 0)$$

The system R_2 is deterministic (i.e., quasi-deterministic and non-subterm-collapsing), but not saturated by paramodulation. There are no right-hand-side critical pairs, so $RHS(R_2) = R_2$. The unification problem is NP-hard by a reduction from monotone 1-in-3-SAT. We only present the key idea here.

For each propositional variable p we form a respective term variable V_p. For each clause $C_i = p_i \vee q_i \vee r_i$ we form the following equation $\mathcal{EQ}_i$.

$$f_1(X_i) =^?_{R_2} g(V_{p_i}, V_{q_i}, V_{r_i})$$

It is not hard to see that the only unifiers of $\mathcal{EQ}_i$ are:

$$\sigma_1 = \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 1,\, X_i \mapsto 0\}$$
$$\sigma_2 = \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 1,\, V_{r_i} \mapsto 0,\, X_i \mapsto s(0)\}$$
$$\sigma_3 = \{V_{p_i} \mapsto 1,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 0,\, X_i \mapsto s(s(0))\}$$

In the case where R is not saturated by paramodulation, there is even a rewrite system whose unification problem is not decidable. Consider the following single-rule rewriting system R_2', where B and $*$ are binary function symbols.

$$B(x, y) * B(u, v) \to B(x * u, y * v)$$

The system R_2' is also deterministic and not saturated by paramodulation. Again, there are no right-hand-side critical pairs, so $RHS(R_2') = R_2'$. The unification problem modulo R_2' is undecidable [1].

3.3 $RHS(R)$ is not Quasi-deterministic

By Lemma 2.1, there are only two ways $RHS(R)$ can be non-quasi-deterministic: either R itself is not quasi-deterministic, or $RHS(R)$ contains a root pair repetition. Furthermore, the only way that R can be non-quasi-deterministic (other than a root pair repetition) is if it contains a root-stable equation. This is because a rewrite rule cannot have a variable as its left-hand side, and a rule with a variable as its right-hand side would make the system subterm-collapsing.

R has a Root-stable Equation

The following system R_3 is non-subterm-collapsing and saturated by paramodulation. However R_3 is not quasi-deterministic because it contains root-stable equations.

$$f(g(0,0,1), f(g(0,1,0), c)) \to f(g(1,0,0), f(g(1,0,0), c))$$

where c, 0, and 1 are constants. Unifiability modulo R_3 can also be shown to be NP-hard by a reduction from monotone 1-in-3-SAT. Again, we only present the key idea here.

For each propositional variable p we form a respective term variable V_p. For each clause $C_i = p_i \vee q_i \vee r_i$ we form the following equation $\mathcal{EQ}_i$, where X_i, Y_i, and Z_i are variables that vary with each clause.

$$f(X_i, f(g(V_{p_i}, V_{q_i}, V_{r_i}), Y_i)) =^?_{R_3} f(g(1,0,0), f(g(1,0,0), Z_i))$$

To unify this equation, we can either unify syntactically, or we can apply the rewrite rule to the left-hand side at either occurrence of f. The following is a complete set of unifiers of $\mathcal{EQ}_i$:

$$\begin{aligned}
\sigma_1 &= \{V_{p_i} \mapsto 1,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 0,\, X_i \mapsto g(1,0,0),\, Y_i \mapsto Z_i\} \\
\sigma_2 &= \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 1,\, V_{r_i} \mapsto 0,\, X_i \mapsto g(0,0,1),\, Y_i \mapsto c,\, Z_i \mapsto c\} \\
\sigma_3 &= \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 1,\, X_i \mapsto g(1,0,0),\, Y_i \mapsto f(g(0,1,0), c), \\
&\qquad Z_i \mapsto f(g(1,0,0), c)\}
\end{aligned}$$

Note that every unifier replaces exactly one of V_{p_i}, V_{q_i}, or V_{r_i} by 1 and the others by 0.

$RHS(R)$ has a Root Pair Repetition

The following system R_4 is non-subterm-collapsing and saturated by paramodulation. However $RHS(R_4)$ is not quasi-deterministic because it has a root pair repetition.

$$\begin{aligned}
f(0,0,1) &\to c_1 & g(0,0,1) &\to c_1 \\
f(0,1,0) &\to c_2 & g(0,1,0) &\to c_2 \\
f(1,0,0) &\to c_3 & g(1,0,0) &\to c_3
\end{aligned}$$

where c_1, c_2, c_3, 0, and 1 are constants. Note that R_4 is deterministic. Unifiability modulo R_4 can also be shown to be NP-hard by a reduction from monotone 1-in-3-SAT. Again, we only present the key idea here.

For each propositional variable p we form a respective term variable V_p. For each clause $C_i = p_i \vee q_i \vee r_i$ we form the following equation $\mathcal{EQ}_i$.

$$f(V_{p_i}, V_{q_i}, V_{r_i}) =^?_{R_4} g(V_{p_i}, V_{q_i}, V_{r_i})$$

The only unifiers of $\mathcal{EQ}_i$ are:

$$\sigma_1 = \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 1\}$$
$$\sigma_2 = \{V_{p_i} \mapsto 0,\, V_{q_i} \mapsto 1,\, V_{r_i} \mapsto 0\}$$
$$\sigma_3 = \{V_{p_i} \mapsto 1,\, V_{q_i} \mapsto 0,\, V_{r_i} \mapsto 0\}$$

4 Undecidability of Subterm-Collapse

For general convergent systems, this result was shown by Bürkert, Herold and Schmidt-Schauß [4]. We show that the property of subterm-collapsing is undecidable even when the convergent system R satisfies the other Lynch-Morawska conditions, namely saturation by paramodulation and quasi-determinism of $RHS(R)$.

The proof is by reduction from the halting problem for reversible deterministic 2-counter Minsky machines, which is undecidable since reversible deterministic 2-counter Minsky machines are Turing-universal [8]. Such a machine M has two counters C_1 and C_2 and is described by a quadruple (Q, δ, q_0, q_L), where $Q = \{q_0, \ldots, q_L\}$ is the set of states, q_0 is the initial state, q_L is the final state, and δ is the move relation. The relation δ is represented by quadruples of the form

$$[q_i, j, s, q_k] \text{ or } [q_i, j, d, q_k]$$

where $q_i, q_k \in Q$, $j = 1$ or 2, $s \in \{Z, P\}$, and $d \in \{-, 0, +\}$. The first form represents checking if counter C_j is zero (Z) or positive (P) and if so moving to state q_k. The second form represents decrementing ($-$), incrementing ($+$), or doing nothing to (0) the counter C_j and then moving to state q_k.

We assume, without loss of generality, that δ does not contain any quadruples with q_0 as the destination. We can assume this because a machine with such a move relation can be transformed into a machine $M' = (Q \uplus \{\, q_0' \,\}, \delta', q_0', q_L)$ where $\delta' = \delta \uplus \{\, [\, q_0', 1, 0, q_0 \,] \,\}$. This new machine is clearly computationally equivalent to M.

We represent the configuration of such machine with a triple (q_i, C_1, C_2), where q_i is the current state of the machine, and C_1 and C_2 are the states (values) of the counters. The machine is assumed to be deterministic and reversible, which means that any configuration has at most one predecessor. By Definition 2.3 of [8], a machine M is deterministic and reversible if and only if, for each pair of quadruples $[\, q_{i_1}, j_1, x_1, q_{i_1'} \,]$ and $[\, q_{i_2}, j_2, x_2, q_{i_2'} \,]$ in δ,

$$(i_1 = i_2 \text{ or } i_1' = i_2') \Rightarrow (j_1 = j_2 \text{ and } \{x_1, x_2\} = \{Z, P\})$$

Thus each state has at most two predecessors and two successors, but only if the state is entered and left (respectively) by checking the state of a single counter.

Given such a machine M and configurations (q_0, k, p) and (q_L, k', p'), we construct a term rewriting system R_M such that R_M is saturated by paramodulation, and $RHS(R_M)$ is quasi-deterministic. We then show that R_M is subterm-collapsing if and only if the machine M, starting in configuration (q_0, k, p), will halt with (q_L, k', p') as its final configuration.

Our system is over the signature $\Sigma = \bigcup_{i=0}^{L} \{\, q_i, f_i, f_i' \,\} \cup \{\, 0, e, e', c, s, f, g, g' \,\}$ where 0, e, e', and $q_0, \ldots, q_L$ are constants, c has arity 4, every other symbol has arity 1.

We can encode a natural number n as a term $s^n(0)$, where s represents the successor function and 0 represents the natural number 0. Each state q_i will encode itself as a constant. We use e and e' to represent the 'empty' state, which means the machine has either not started or has already halted.

We can then encode a configuration (q_i, k, p) as a term $c(q_i, s^k(0), s^p(0), s^n(0))$, where n is the number of steps the machine has taken. The symbol c acts only as a constructor. We use the f and g symbols, with various subscripts and primes, to ensure termination and saturation by paramodulation of the resulting rewrite system.

To start, initialize R_M to the following rewrite system:

$$\begin{aligned} f(c(e, s^k(0), s^p(0), 0)) &\to c(q_0, s^k(0), s^p(0), 0) \\ f_L(c(q_L, s^{k'}(0), s^{p'}(0), z)) &\to g(c(e', 0, 0, z)) \\ g'(g(c(e', 0, 0, s(z)))) &\to c(e', 0, 0, z) \\ g'(g(c(e', 0, 0, 0))) &\to e \end{aligned}$$

The first rule encodes initializing the machine, moving from an empty state e to the initial state q_0 iff the configuration matches the given starting configuration. The second rule terminates the machine iff the configuration matches the given final configuration and moves to an empty state e'. The third rule checks that the fourth argument in the c-term encodes a valid natural number. The last rule takes a correctly terminated machine and gives the empty state, which could cause a subterm-collapse.

Next we encode the transition rules. For each quadruple in δ, extend R_M using one of the following transformations:

(a1) $[\, q_i, 1, P, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, s(x), y, z)) \to c(q_j, s(x), y, s(z)) \,\}$
(a2) $[\, q_i, 2, P, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, x, s(y), z)) \to c(q_j, x, s(y), s(z)) \,\}$

(b1) $[\, q_i, 1, Z, q_j \,]\colon R_M := R_M \cup \{\, f_i'(c(q_i, 0, y, z)) \to c(q_j, 0, y, s(z)) \,\}$
(b2) $[\, q_i, 2, Z, q_j \,]\colon R_M := R_M \cup \{\, f_i'(c(q_i, x, 0, z)) \to c(q_j, x, 0, s(z)) \,\}$

(c1) $[\, q_i, 1, +, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, x, y, z)) \to c(q_j, s(x), y, s(z)) \,\}$
(c2) $[\, q_i, 2, +, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, x, y, z)) \to c(q_j, x, s(y), s(z)) \,\}$

(d1) $[\, q_i, 1, 0, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, x, y, z)) \to c(q_j, x, y, s(z)) \,\}$
(d2) $[\, q_i, 2, 0, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, x, y, z)) \to c(q_j, x, y, s(z)) \,\}$

(e1) $[\, q_i, 1, -, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, s(x), y, z)) \to c(q_j, x, y, s(z)) \,\}$
(e2) $[\, q_i, 2, -, q_j \,]\colon R_M := R_M \cup \{\, f_i(c(q_i, x, s(y), z)) \to c(q_j, x, y, s(z)) \,\}$

Lemma 4.1. *Given a reversible deterministic 2-counter Minsky machine M, the system R_M has no RHS overlaps.*

Proof. Suppose there is an RHS overlap between two rules in R_M. The only way for this to happen is for the rules' right hand sides to both be c-terms and to contain the same state q_i. Because M is a reversible deterministic machine, and because we assume there is no quadruple with q_0 as its destination, the only way for that to happen is for one c-term to have been created by either transformation (a1) or (a2) (i.e., the quadruple in δ contained a P) and the other by the corresponding (b1) or (b2) (i.e., a Z). However, in that case, there can be no overlap, because there will be a function clash between s and 0. Thus there are no RHS overlaps. □

Lemma 4.2. *Given a reversible deterministic 2-counter Minsky machine M, the system R_M is saturated by paramodulation and quasi-deterministic.*

Proof. That R_M is saturated by paramodulation follows from the fact that none of the root symbols of the left hand sides of rules in R_M occur at a position in the left- or right-hand side of any other rule. Therefore, there are no possible overlaps and R_M is saturated trivially.

Recall that for the system R_M to be quasi-deterministic, the following conditions must hold:

(1) R_M has no root pair repetitions
(2) If $(s \to t) \in R_M$, then neither t nor s is a variable
(3) If $(s \to t) \in R_M$, then $root(s) \neq root(t)$

For each quadruple in δ, only one of the above transformations will be used to extend R_M (i.e., only one rule will be added). The only way two rules could have the same left-hand root symbol is for them to contain the same state q_i, and that is only possible if one was created by either transformation (a1) or (a2) (i.e., the quadruple in δ contained a P) and the other by the corresponding (b1) or (b2) (i.e., a Z). In that case, however, one would have f_i as its left-hand root symbol, and the other would have f_i'. Thus condition 1 is met.

Conditions 2 and 3 are readily apparent from the rules in R_M. □

Lemma 4.3. *Given a reversible deterministic 2-counter Minsky machine M, the system R_M is convergent.*

Proof. To prove that R_M is terminating, consider the measure function

$$\phi\colon T(\Sigma, \mathcal{X}) \to \mathbb{N} \times \mathbb{N}$$

such that $\phi(t) = (n_1, n_2)$, where n_1 is the number of occurrences of f, f_i, and f_i' symbols for all i, and n_2 is the number of occurrences of g and g' symbols. Note that every rule *except*

$$f_L(c(q_L, s^{k'}(0), s^{p'}(0), z)) \to g(c(e', 0, 0, z))$$

removes at least one of these symbols and does not add any of the others. For this remaining rule, the first component decreases even though the second component increases. Since the rules in R_M are linear and variable-preserving, if $t_1 \to_{R_M} t_2$, then $\phi(t_1) >_{lex} \phi(t_2)$, where $>_{lex}$ is the lexicographic order induced by $>$ on $\mathbb{N} \times \mathbb{N}$. Therefore R_M is terminating.

We will show the confluence of R_M by showing that R_M has no critical pairs. As shown in Lemma 4.2, no two rules added by transformations (a1)–(e2) have the same root symbol on their left-hand sides. No rule's left-hand side's root symbol could occur at a non-root position of another rule's left-hand side. So the only possible overlap would be the third and fourth initial rules, but note that their left-hand sides are not unifiable due to a function clash of 0 and $s(z)$. Thus R_M has no critical pairs. So R_M is convergent. □

Lemma 4.4. *Given a reversible deterministic 2-counter Minsky machine M, configurations (q_0, k, p) and (q_L, k', p'), and rewrite system R_M constructed as above, R_M is subterm-collapsing if and only if the machine M, starting in configuration (q_0, k, p), will halt with (q_L, k', p') as its final configuration.*

Proof of 'if'. Suppose the machine M, starting in the given initial configuration, does halt with the given final configuration after n steps. Since M is deterministic, there is a single sequence of transitions the machine can take. Let $S = \langle \tau_1, \ldots, \tau_n \rangle$ be that sequence of transitions. For each $\tau_i = [q_j, \kappa, d, q_{j'}]$, we define the symbol f_i^* as

$$f_i^* = \begin{cases} f_j' & \text{if } d = Z \\ f_j & \text{otherwise} \end{cases}$$

In other words, f_i^* is an alias for either f'_j or f_j depending on how M left state q_j.

Now we can construct a term $t = f_L((f_n^* \circ \cdots \circ f_1^*)(f(c(e, s^k(0), s^p(0), 0))))$, where $\circ$ is function composition, such that $t \to^+_{R_M} g(c(e', 0, 0, s^n(0)))$. This should be apparent from the definition of R_M. After the first rule initializes the machine and removes the f, each rewrite will remove the innermost f_i^* and move to the next configuration. Finally the second rule will remove the outermost f_L and move to the empty state.

Finally we can create a term $t' = (g' \circ g)^n(g'(t))$ such that $t' \to^+_{R_M} e$. This is because each application of the third rule will remove the innermost $(g' \circ g)$ and an s from the step counter, until the counter becomes 0. At that point, the fourth rule will rewrite the entire term to e. Note that e is a proper subterm of t, which is itself a subterm of t', so R_M is subterm-collapsing. □

Proof of 'only if'. Conversely, suppose R_M is subterm-collapsing. Since all rewrite rules in R_M are variable-preserving, and each rule added by transformations (a1)–(e2) increases the step counter in the c-term, each application of a rule creates a c-term which has never occurred before. Therefore, the only way for a subterm-collapse to occur is for a term to rewrite to e by the fourth rule.

Let t be a term with e as a proper subterm such that $t \to^+_{R_M} e$. The only way to rewrite t is by the first rule. This means there is a position p in t such that $t|_p = f(c(e, s^k(0), s^p(0), 0))$. Therefore, there is a term t' such that $t \to_{R_M} t'$ and $t'|_p = c(q_0, s^k(0), s^p(0), 0)$.

At this point, the only way to rewrite t' is to use the rules added by transformations (a1)–(e2). Therefore, there must be a sequence $S = \langle i_1, \ldots, i_n \rangle$, $0 \leq i_j < L$, and a position p' in t, $p' < p$, such that $t|_{p'} = t'|_{p'} = (f_{i_n}^* \circ \cdots \circ f_{i_1}^*)(t|_p)$, where each f_j^* is either f_j or f'_j. Therefore, there is a term t'' such that $t \to_{R_M} t' \to^+_{R_M} t''$ and $t''|_{p'} = c(q_L, s^{k'}(0), s^{p'}(0), s^n(0))$. It should be clear at this point, due to the way the rules of R_M are constructed, that if the machine M is started in configuration (q_0, k, p), it would move through the states indexed by the sequence S, finally halting in configuration (q_L, k', p'). □

Theorem 4.5. *It is undecidable, given a rewrite system R which is convergent, saturated by paramodulation, and for which $RHS(R)$ is quasi-deterministic, whether $RHS(R)$ is subterm-collapsing.*

Proof. As shown in Lemma 4.4, the halting problem for reversible deterministic 2-counter Minsky machines can be reduced to checking for subterm-collapse in a rewrite system R_M. Since R_M has no RHS overlaps (Lemma 4.1), $RHS(R_M) = R_M$. By Lemma 4.2, R_M is saturated by paramodulation and quasi-deterministic, and by Lemma 4.3, R_M is convergent. Therefore, since the halting problem for reversible deterministic 2-counter Minsky machines is undecidable, so is checking for subterm-collapse in such a rewrite system. □

5 Conclusion and Future Work

In this paper, we provesd that relaxing any of the Lynch-Morawska conditions can give rise to unification problems that are in NP (or are not even decidable!). In doing so, we answered an open problem posed in the Lynch-Morawska paper [7]. We also extended previous work [4] on undecidability of subterm-collapse to the (very) restricted term rewriting systems considered in this paper. So while the Basic Syntactic Mutation algorithm guarantees a polynomial time unification algorithm for term rewriting systems that fit its conditions, determining whether a system has these properties cannot be automated.

Our future work involves extending the unification algorithm given in [7] to solve the asymmetric unification problem for theories which satisfy the Lynch-Morawska conditions. This type of unification problem has only recently been studied and has applications in cryptographic protocol analysis [5].

References

[1] S. Anantharaman, S. Erbatur, C. Lynch, P. Narendran, and M. Rusinowitch. Unification Modulo Synchronous Distributivity. In B. Gramlich, D. Miller, and U. Sattler, editors, *IJCAR*, volume 7364 of *Lecture Notes in Computer Science*, pages 14–29. Springer, 2012.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1999.

[3] F. Baader and W. Snyder. Unification Theory. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 440–526. Elsevier Science Publishers BV, 1999.

[4] H.J. Bürckert, A. Herold, and M. Schmidt-Schauss. On Equational Theories, Unification, and (Un)Decidability. *Journal of Symbolic Computation*, 8(1):3–49, 1989.

[5] S. Erbatur, S. Escobar, D. Kapur, Z. Liu, C. Lynch, C. Meadows, J. Meseguer, P. Narendran, S. Santiago, and R. Sasse. Effective Symbolic Protocol Analysis via Equational Irreducibility Conditions. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS*, volume 7459 of *Lecture Notes in Computer Science*, pages 73–90. Springer, 2012.

[6] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[7] C. Lynch and B. Morawska. Basic Syntactic Mutation. In A. Voronkov, editor, *CADE*, volume 2392 of *Lecture Notes in Computer Science*, pages 471–485. Springer, 2002.

[8] K. Morita. Universality of a reversible two-counter machine. *Theoretical Computer Science*, 168(2):303–320, 1996.

[9] E. Ohlebusch. Termination is not Modular for Confluent Variable-Preserving Term Rewriting Systems. *Inf. Process. Lett.*, 53(4):223–228, 1995.

A Forward Closure

In an earlier version of the paper, we formulated the Lynch-Morawska conditions in terms of the concept of *forward closure*. The forward closure of a convergent term rewriting system R is defined in terms of the following operation on rules in R: Let $\rho_1 = (l_1 \to r_1)$ and $\rho_2 = (l_2 \to r_2)$ be two rules in R and let $p \in \mathcal{FP}os(r_1)$. Then

$$\rho_1 \rightsquigarrow_p \rho_2 = \sigma(l_1 \to r_1[r_2]_p)$$

where $\sigma = mgu(r_1|_p =^? l_2)$.

Given rewrite systems R_1, R_2, and R_3 we define $\mathcal{FOV}(R_1, R_2)$ (forward overlap) and $\mathcal{N}(R_1, R_2, R_3)$ (non-redundant) as

$$\mathcal{FOV}(R_1, R_2) = \{\, \rho_1 \rightsquigarrow_p \rho_2 \mid \rho_1 = (l_1 \to r_1) \in R_1,\ \rho_2 \in R_2, \text{ and } p \in \mathcal{FP}os(r_1) \,\}$$
$$\mathcal{N}(R_1, R_2, R_3) = \{\, \rho \mid \rho \in \mathcal{FOV}(R_1, R_2) \text{ and } \rho \text{ is not redundant in } R_3 \,\}$$

We now simultaneously define $NR_k(R)$ (new rules) and $FC_k(R)$ (forward closure) for all $k \geq 0$.

$$NR_0(R) = R \qquad NR_{k+1}(R) = \mathcal{N}(NR_k(R), R, FC_k(R))$$
$$FC_0(R) = R \qquad FC_{k+1}(R) = FC_k(R) \cup \mathcal{N}(NR_k(R), R, FC_k(R))$$

Finally,

$$FC(R) = \bigcup_{i=1}^{\infty} FC_i(R)$$

Note that $FC_j(R) \subseteq FC_{j+1}(R)$ for all $j \geq 0$. A set of rewrite rules R is *forward-closed* if and only if $FC(R) = R$.

However it turns out that:

Lemma A.1. *It is not the case that every convergent, forward-closed term rewriting system is saturated by paramodulation.*

Proof. Consider the following term rewriting system R:

$$f(g(x,x)) \to h(x,x)$$
$$g(s(x), s(y)) \to h(s(x), s(y))$$
$$f(h(x,x)) \to h(x,x)$$

The only nontrivial critical pair is between the first and second rules. This pair is given as follows:

$$\langle f(h(s(u), s(v))), h(s(u), s(v)) \rangle$$

The equation formed by this pair, $f(h(s(u), s(v))) \approx h(s(u), s(v))$, is *not* redundant, since $h(s(u), s(v))$ is in normal form and the only way to join the terms is by the third rule. The instance $f(h(s(u), s(v))) \to h(s(u), s(v))$ is equal to the equation, and thus not lower in the ordering. □