Basic Syntactic Mutation

Christopher Lynch and Barbara Morawska

Department of Mathematics and Computer Science Box 5815, Clarkson University, Potsdam, NY 13699-5815, USA, E-mail:
clynch@clarkson.edu,morawskb@clarkson.edu **

Abstract. We give a set of inference rules for E-unification, similar to the inference rules for Syntactic Mutation. If the E is finitely saturated by paramodulation, then we can block certain terms from further inferences. Therefore, E-unification is decidable in NP, as is also the case for Basic Narrowing. However, if we further restrict E, then our algorithm runs in quadratic time, whereas Basic Narrowing does not become polynomial, since it is still nondeterministic.

1 Introduction

E-unification is the problem of deciding if there are substitutions for variables which make two terms equal modulo an equational theory E. E-unification occurs in many applications. Unfortunately, it is an undecidable problem in general. We are interested in finding classes of equational theories where the E-unification problem is decidable and tractable.

One method of attacking this problem is to examine equational theories which are finitely saturated under a given set of inference rules. For example, if an equational theory E is saturated under the Critical Pair rule of Knuth-Bendix Completion[11], then the word problem is decidable for E, i.e., the problem of deciding if two terms are equal modulo E. However, the E-unification problem can still be undecidable for such theories.

The Critical Pair rule allows inferences only into a subterm of the larger side of an equation. It can be extended to an inference rule called Paramodulation[3], which allows inferences also into the smaller side of an equation. In [14], it is shown that if E is saturated by Paramodulation then the E-unification problem is decidable, and furthermore the decision procedure is in NP. In the Narrowing procedure used in that paper, whenever Narrowing is performed, the smaller side of the equation from E is marked in the conclusion and future inferences are not allowed into the marked positions. Each inference "consumes" a position of the goal, and therefore each Narrowing sequence halts in a linear number of steps. Therefore, the procedure is in NP, since it is a non-deterministic procedure.

Here, we also consider equational theories E saturated under Paramodulation. The inference system we use is not Narrowing, but a variant of the Syntactic

** This work was supported by NSF grant number CCR-0098270 and ONR grant number N00014-01-1-0435.

Mutation inference rules of [10]. However, our inference rules are Basic, in the sense that we can mark terms from the equation from E, and not allow any more inferences into these terms. Therefore, just like in [14] we get an NP algorithm.

The important part of our paper is what comes next. We show that if E is further restricted (see section 6), then the algorithm is no longer nondeterministic. In fact, the algorithm runs in a linear number of steps, in $O(n^2)$ time. This is in contrast to Basic Narrowing, where these restrictions do not allow the procedure to become deterministic, so it does not become polynomial.[1]

We show an interesting connection to Syntactic Theories[10]. If E' is saturated by Paramodulation, then we can always quickly perform a few extra inference rules to E', yielding E. Then E is a resolvent presentation of a syntactic theory. Our results basically follow from the fact that E is resolvent, and there is an equivalent subset of E such that all proper subterms in E are reduced by E.

Most of this paper deals with the set of inference rules yielding the NP algorithm. The inference rules have been designed so that when we present the definition of restricted equations, the polynomial time result is almost immediate.

Our full proofs are given in [13].

2 Preliminaries

We assume standard definitions of term rewriting[1].

We assume we are given a set of variables and a set of uninterpreted function symbols of various arities. *Terms* are defined recursively in the following way: each variable is a term, and if $t_1, \cdots, t_n$ are terms, and f is of arity $n \geq 0$, then $f(t_1, \cdots, t_n)$ is a term, and f is the symbol at the *root* of $f(t_1, \cdots, t_n)$. A term (or any object) without variables is called *ground*. We consider equations of the form $s \approx t$, where s and t are terms. Let E be a set of equations, and $u \approx v$ be an equation, then we write $E \models u \approx v$ (or $u =_E v$) if $u \approx v$ is true in any model of E. If G is a set of equations, then $E \models G$ if and only if $E \models e$ for all e in G.

A *substitution* is a mapping from the set of variables to the set of terms, such that it is almost everywhere the identity. We identify a substitution with its homomorphic extension. If θ is a substitution then $Dom(\theta) = \{x \mid x\theta \neq x\}$ and $Range(\theta) = \{x\theta \mid x \in Dom(\theta)\}$. If R_E is a set of rewrite rules, then a substitution θ is R_E-reduced if all terms in $Range(\theta)$ are R_E-reduced.

A substitution θ is an *E-unifier* of an equation $u \approx v$ if $E \models u\theta \approx v\theta$. θ is an *E-unifier* of a set of equations G if θ is an E-unifier of all equations in G.

If σ and θ are substitutions, then we write $\sigma \leq_E \theta[Var(G)]$ if there is a substitution ρ such that $E \models x\sigma\rho \approx x\theta$ for all x appearing in G. If G is a set of equations, then a substitution θ is a *most general E-unifier of G*, written $\theta = mgu(G)$ if θ is an E-unifier of G, and for all E-unifiers σ of G, $\theta \leq_E \sigma[Var(G)]$. A complete set of E-unifiers of G, is a set of E-unifiers Θ of G such that for all E-unifiers σ of G, there is a θ in Θ such that $\theta \leq_E \sigma[Var(G)]$.

[1] See the example in Section 7.

Given a unification problem we can either solve the unification problem or decide the unification problem. Given a goal G and a set of equations E, to *solve* the unification problem means to find a complete set of E-unifiers of G. To *decide* the unification problem simply means to answer true or false as to whether G has an E-unifier.

If E is a set of equations, then define $Gr(E)$ as the set of all ground instances of equations in E. We assume a reduction ordering $\prec$ on E, which is total on ground terms. In order to extend the ordering to equations, we treat equations as multisets of terms, i.e. $(s \approx t) \prec (u \approx v)$ iff $\{s,t\} \prec_{mul} \{u,v\}$.

3 Saturation

We will show that if E is a finite set of equations *saturated* by *Paramodulation*, then the E-unification problem is in NP. Paramodulation and saturation are defined below.

Paramodulation

$$\frac{u[s'] \approx v \quad s \approx t}{u[t]\sigma \approx v\sigma}$$

where $\sigma = mgu_{\emptyset}(s,s')$, $s\sigma \not\preceq t\sigma$, and s' is not a variable.

This inference rule is an extension of the Critical Pair rule, which also allows inferences into the smaller side of an equation.

In a set E of ground equations, an inference is *redundant* if its conclusion follows from equations of E smaller than its largest premise. In a general set of equations E, an inference is *redundant* if it is redundant in $Gr(E)$. A set of equations is *saturated* if all of the inferences among equations in E are redundant. Automated theorem provers generally saturate a set of equations by some inference rule.

In this section we will inductively define a set R_E of rewrite rules from an equational theory E. This construction is originally from [2]. R_E will be used in the completeness proof of the inference system we give in the next section. A rule $s \to t$ is *reducible* by some set of rules T (T*-reducible*), if there is a rule $u \to v \in T$ different from $s \to t$ such that u is a subterm of s or t.

Definition 1. *For each $s \approx t \in Gr(E)$ such that $s \succ t$,*

- $I^{s\approx t} = \begin{cases} \emptyset, & \textit{if } s \textit{ or } t \textit{ is reducible by } R^{\prec s\approx t} \\ \{s \to t\}, & \textit{otherwise.} \end{cases}$
- $R^{\prec s\approx t} = \bigcup_{(u\approx v)\prec(s\approx t)} I^{u\approx v}$
- $R_E = \bigcup_{s\approx t \in Gr(E)} I^{s\approx t}$

Proposition 1. *The term rewriting system R_E is confluent and terminating.*

Lemma 1. *If $s \approx t$ is in $Gr(E)$ and $s \to t$ is R_E-reducible, then $s \to t$ is $R^{\prec s \approx t}$-reducible.*

Corollary 1. *If $s \to t$ is R_E-reducible and $s \approx t \in Gr(E)$, then $s \to t \notin R_E$.*

The corollary follows, because if $s \to t$ is R_E-reducible, then it is $R^{\prec s \approx t}$-reducible, and hence $I^{s \approx t} = \emptyset$. Therefore $s \to t \notin R_E$.

R_E^* denotes a congruence induced by R_E.

Theorem 1. *If E is saturated under Paramodulation, $s \approx t \in E$ and σ is a ground substitution, then $R_E^* \models s\sigma \approx t\sigma$.*

4 The *BSM* Algorithm

In this section we give an algorithm for E-unification. It is based on a set of inference rules and a selection rule. The algorithm is "don't know" non-deterministic, i.e. sometimes more than one inference rule has to be checked. Because we assume that all applicable equations will be used in inference rules, and since R_E is logically equivalent to E, we can assume in our completeness proof in the ground case that equations used are from R_E. Therefore, the proper subterms will be reduced by R_E, hence we can argue that no inferences will need to take place in those terms. Therefore, we will forbid inferences into them. This will restrict the search space, and allow us to show that the algorithm will halt. The terms of which we assume that their ground instances are reduced will be marked with boxes.

We define the Right-Hand-Side Critical Pair rule:

Right-Hand-Side Critical Pair (at the root)

$$\frac{s \approx t \quad u \approx v}{s\sigma \approx u\sigma}$$

where $s\sigma \not\prec t\sigma$, $u\sigma \not\prec v\sigma$, $\sigma \approx mgu_{\emptyset}(v,t)$ and $s\sigma \neq u\sigma$.

Define $RHS(E) = \{e \mid e$ is the conclusion of a Right-Hand-Side Critical Pair inference of two members of $E\} \cup E$. This is not a saturation, because conclusions of these inferences cannot be used in further inferences with Right-Hand-Side Critical Pair rule . Therefore, $RHS(E)$ can be computed in quadratic time and only adds a quadratic number of equations to E.

Note that, if $s\sigma\gamma \to t\sigma\gamma$ and $u\sigma\gamma \to v\sigma\gamma$, for some ground substitution γ, are in R_E, then all proper subterms in the equation $s\sigma\gamma \approx u\sigma\gamma$ are R_E-reduced. We will show that if E is saturated under Paramodulation, then $RHS(E)$ is a Syntactic Theory. This allow us to design a decision procedure for E-unification.

Theorem 2. *Let $E = RHS(E')$, where E' is finite and saturated by Paramodulation. Then, for each ground R_E-reduced equation $u \approx v$, such that $E \models u \approx v$ one of the following is true:*

1. $u = f(u_1, \ldots, u_n)$, $v = f(v_1, \ldots, v_n)$ *and* $E \models \bigcup_i^n u_i \approx v_i$
2. $u = f(u_1, \ldots, u_n)$, $v = g(v_1, \ldots, v_m)$ *and there is* $f(s_1, \ldots, s_n) \approx t \in E$ *and an* R_E*-reduced substitution* σ*, such that* $E \models \bigcup_i^n u_i \approx s_i\sigma$ *and* $E \models g(v_1, \ldots, v_m) \approx t\sigma$*, and if* $t = g(t_1, \ldots, t_m)$*, then* $E \models \bigcup_j^m v_j \approx t_j\sigma$*. All* $s_i\sigma$*,* $t_j\sigma$ *are* R_E*-reduced.*

Proof. $E \models u \approx v$ and $u \approx v$ is a ground equation, hence also $R_E^* \models u \approx v$ (by Theorem 1). Consider the rewrite proof in R_E of $u \approx v$. All RHS of the rules in R_E are R_E-reduced, so in a rewrite proof of $t \xrightarrow{*} t'$ in R_E, for ground terms t and t', where t' is the normal form of t, there may be some steps reducing subterms of t and then at most one step at the root at the end, reducing the whole term to t'.

$$t = f(t_1, \ldots, t_n) \xrightarrow{*} f(t'_1, \ldots, t'_n) \overset{\text{root-step}}{\rightarrow} t'$$

Hence we have 3 cases here:

i. No step at the root of either side in the proof of $u \approx v$:

$$u = f(u_1, \ldots, u_n) \xrightarrow{*} f(u'_1, \ldots, u'_n) \xleftarrow{*} f(v_1, \ldots, v_n) = v$$

Then $R_E \models u_i \xrightarrow{*} u'_i$ for all $i = 1, \ldots, n$. Hence also $E \models u_i \approx u'_i$. $v_j \xrightarrow{*} u'_j \in R$ for all $j = 1, \ldots, n$. Hence also $E \models v_j \approx u'_j$. Hence $E \models u_j \approx v_j$, for all $j = 1, \ldots, n$ and the first statement of the theorem is true.

ii. One step at the root in the proof of $u \approx v$. Then $u = f(u_1, \ldots, u_n)$ and $v = g(v_1, \ldots, v_m)$. Suppose that there is a step at the root in the proof:

$$f(u_1, \ldots, u_n) \xrightarrow{*} f(u'_1, \ldots, u'_n) \overset{\text{root-step}}{\rightarrow} u'$$

where u' is an R_E-normal form of u, and there is no step at the root in the proof:

$$g(v_1, \ldots, v_m) \xrightarrow{*} u'$$

Hence $u' = g(v'_1, \ldots, v'_m)$ and the step at the root has the form: $f(u'_1, \ldots, u'_n) \rightarrow g(v'_1, \ldots, v'_m)$. Hence this must be a rule in R_E. Therefore there are two possibilities:

(a) $f(s_1, \ldots, s_n) \approx g(t_1, \ldots, t_m) \in E$ and there is a R_E-reduced substitution σ, such that $s_i\sigma = u'_i$, for all $i = 1, \ldots, n$, and $t_j\sigma = v'_j$, for all $j = 1, \ldots, m$. Since $u_i \xrightarrow{*} u'_i$, for all $i = 1, \ldots, n$, $E \models \bigcup_i^n u_i \approx s_i\sigma$.
Since $R_E \models g(v_1, \ldots, v_m) \xrightarrow{*} g(v'_1, \ldots, v'_m)$, then $E \models g(v_1, \ldots, v_m) \approx g(t_1, \ldots, t_m)\sigma$ and since $v_j \xrightarrow{*} v'_j$ for all $j = 1, \ldots, m$, $E \models \bigcup_j^m v_j \approx t_j\sigma$,

(b) $f(s_1, \ldots, s_n) \approx x \in E$, and there is a R_E-reduced substitution σ, such that $s_i\sigma = u'_i$, for all $i = 1, \ldots, n$, and $x\sigma = g(v'_1, \ldots, v'_m)$.
Since $u_i \xrightarrow{*} u'_i$, for all $i = 1, \ldots, n$, $E \models \bigcup_i^n u_i \approx s_i\sigma$. Since $R_E \models g(v_1, \ldots, v_m) \xrightarrow{*} g(v'_1, \ldots, v'_m)$, then $E \models g(v_1, \ldots, v_m) \approx x\sigma$.

Since $f(u'_1,\dots,u'_n) \rightarrow g(v'_1,\dots,v'_m)$ is in R_E, all subterms u'_i and v'_j are R_E-reduced. Hence the second statement of the theorem is true.

iii. Two steps at the root in the proof of $u \approx v$. Then $u = f(u_1,\dots,u_n)$ and $v = g(v_1,\dots,v_m)$. The rewrite proof has the following form:

$$f(u_1,\dots,u_n) \overset{*}{\rightarrow} f(u'_1,\dots,u'_n) \overset{\text{root-step}}{\rightarrow} w \overset{\text{root-step}}{\leftarrow} g(v'_1,\dots,v'_m) \overset{*}{\leftarrow} g(v_1,\dots,v_m)$$

where w is a normal form for both terms and $f(u'_1,\dots,u'_n) \neq g(v'_1,\dots,v'_m)$. (The case where $f(u'_1,\dots,u'_n) = g(v'_1,\dots,v'_m)$ reduces to the first case, since there is a proof of $u \approx v$ with no step at the root on either side.) Since $R_E \models u_i \overset{*}{\rightarrow} u'_i$ for each $i = 1,\dots,n$, $E \models \bigcup_i^n u_i \approx u'_i$ and since $R_E \models g(v_1,\dots,v_m) \overset{*}{\rightarrow} g(v'_1,\dots,v'_m)$, $E \models g(v_1,\dots,v_m) \approx g(v'_1,\dots,v'_m)$. The subterms u'_i and v'_j are all R_E-reduced.
Since $f(u'_1,\dots,u'_n) \rightarrow w$ and $g(v'_1,\dots,v'_m) \rightarrow w$ are in R_E, hence there must be an equation $f(s_1,\dots,s_n) \approx t$ in E and also $g(t_1,\dots,t_m) \approx t'$ in E, and an R_E-reduced substitution σ, such that $s_i\sigma = u'_i$, for all $i = 1,\dots,n$, and also $t_j\sigma = v'_j$, for all $j = 1,\dots,m$, and $t\sigma = w$ and $t'\sigma = w$. By the saturation with the Right-Hand-Side Critical Pair rule, also $f(s_1,\dots,s_n)\theta \approx g(t_1,\dots,t_m)\theta$ is in E, where $\theta = mgu_\emptyset(t,t')$. Obviously, $\theta \leq \sigma$, and hence for some τ, $s_i\theta\tau = s_i\sigma$ for any $i = 1,\dots,n$ and $t_j\theta\tau = t_j\sigma$ for each $j = 1,\dots,m$. The second statement of the theorem is therefore true.

Our inference rules are presented in Figures 1 and 2. They use a *selection rule* which is defined after the inference rules. We call the set of inference rules BSM (Basic Syntactic Mutation). We also define a procedure called BSM, which is the result of closing a set of equations under the inference rules BSM.

We treat the equations in the inference rules as symmetric, i.e., an equation $s \approx t$ can also be viewed as $t \approx s$.

The boxed elements in the assumptions of the rules are boxed also in the conclusion. The subterms of boxed terms are treated as also boxed. In the inference rules, if we do not box a term then it can be either boxed or unboxed, unless we explicitly say that it is not boxed.

The rule Imitation is allowed only when there are multiple equations with variable x on one side and terms with the same function symbol on the other side, as in the example:

$$\frac{\{x \approx f(a), x \approx f(b), x \approx f(c)\} \cup G}{\{x \approx \boxed{f(y)}, y \approx a, y \approx b, y \approx c\} \cup G}$$

In the case where all function symbols are different, Imitation is not applicable, instead we must use Mutate&Imitate in a successful proof, as in the example:

$$\frac{\{x \approx f(a), x \approx g(b)\} \cup G}{\{x \approx g(y), \boxed{b} \approx b, y \approx \boxed{b}, \boxed{a} \approx a\} \cup G}$$

where $f(a) \approx g(b)$ is in E.

Decomposition:

$$\frac{\{f(u_1,\cdots,u_n) \approx f(v_1,\cdots,v_n)\} \cup G}{\{u_1 \approx v_1,\cdots,u_n \approx v_n\} \cup G}$$

where $f(u_1,\cdots,u_n) \approx f(v_1,\cdots,v_n)$ is selected.

Mutate:

$$\frac{\{f(u_1,\cdots,u_n) \approx g(v_1,\cdots,v_m)\} \cup G}{\bigcup_i\{u_i \approx \boxed{s_i}\} \cup \bigcup_i\{\boxed{t_i} \approx v_i\} \cup G}$$

where $f(u_1,\cdots,u_n) \approx g(v_1,\cdots,v_m)$ is selected, $f(u_1,\cdots,u_n)$ is not boxed and $f(s_1,\cdots,s_n) \approx g(t_1,\cdots,t_m) \in E$.

Imitation:

$$\frac{\bigcup_i\{x \approx f(v_{i_1},\cdots,v_{i_n})\} \cup G}{\{x \approx \boxed{f(y_1,\cdots,y_n)}, \bigcup_i\{y_1 \approx v_{i_1}\},\cdots,\bigcup_i\{y_n \approx v_{i_n}\}\} \cup G}$$

where $i > 1$ and at least two of $\bigcup_i\{x \approx f(v_{i_1},\cdots,v_{i_n})\}$ are selected, and there are no more equations of the form $x \approx f(u_1,\cdots,u_n)$ in G.

Mutate&Imitate:

$$\frac{\{x \approx f(u_1,\cdots,u_n), x \approx g(v_1,\cdots,v_m)\} \cup G}{\begin{array}{c}\{x \approx f(y_1,\cdots,y_m), y_1 \approx \boxed{s_1},\ldots,y_n \approx \boxed{s_n}, \boxed{s_1} \approx u_1,\cdots,\boxed{s_n} \approx u_n, \\ \boxed{t_1} \approx v_1,\cdots,\boxed{t_m} \approx v_m\} \cup G\end{array}}$$

where $f(s_1,\cdots,s_n) \approx g(t_1,\cdots,t_m)$ is in E, $x \approx f(u_1,\cdots,u_n)$ and $x \approx g(v_1,\cdots,v_m)$ are selected in the goal and

1. $f(u_1,\cdots,u_n)$ is boxed , $g(v_1,\cdots,v_m)$ is unboxed in the premise and $f(y_1,\cdots,y_m)$ is boxed in the conclusion, or
2. both $f(u_1,\cdots,u_n)$ and $g(v_1,\cdots,v_m)$ are not boxed in the premise and $f(y_1,\cdots,y_m)$ is not boxed in the conclusion.

Variable Elimination:

if $x \not\approx y$:	otherwise:
$$\frac{x \approx y, \quad G}{x \approx y, \quad G[x \mapsto y]}$$	$$\frac{x \approx x \cup G}{G}$$
where both x and y appear in G.	

Fig. 1. The BSM inference rules

VariableMutate:

$$\frac{\{f(u_1, \cdots, u_n) \approx v\} \cup G}{\{u_1 \approx \boxed{s_1}, \cdots, u_n \approx \boxed{s_n}, \quad x \approx v\} \cup G}$$

where $f(u_1, \cdots, u_n) \approx v$ is selected, $f(u_1, \cdots, u_n)$ is not boxed and there is an equation of the form $f(s_1, \cdots, s_n) \approx x \in E$.

Mutate&Imitate-cycle:

$$\frac{\{x \approx f(v_1, \ldots, v_n)\} \cup G}{\{x \approx \boxed{g(t_1, \ldots, t_k)}, \bigcup_i^n \boxed{s_i} \approx v_i\} \cup G}$$

where $g(t_1, \ldots, t_k) \approx f(s_1, \ldots, s_n) \in E$, $x \approx f(v_1, \ldots, v_n)$ only is selected.

Imitation-cycle:

$$\frac{\{x \approx f(v_1, \cdots, v_n)\} \cup G}{\{x \approx \boxed{f(y_1, \cdots, y_n)}, y_1 \approx v_1, \cdots, y_n \approx v_n\} \cup G}$$

where $x \approx f(v_1, \cdots, v_n)$ only is selected.

Fig. 2. *BSM* inference rules continued

Definition 2. *We recursively define an equation $x \approx t$ in G to be* solved *if the variable x does not appear in $G \backslash \{x \approx t\}$ or the variable x does not appear in an unsolved equation in G. The variable x is then called* solved.

We use a notion of cycle in the definition of our selection rule. By *cycle*, we understand a set of equations of the type $x \approx t$, where x is a variable, t is a term, that can be ordered as $\{x_1 = t_1, \ldots, x_n = t_n\}$, in such a way that $\{x_{i+1}\} \cap Var(t_i) \neq \emptyset$, where at least one t_i is not variable and $x_1 \in Var(t_n)$.

The following selection rule is used in the inference rules. Conditions imposed by the definition deal with "don't care" nondeterminism in the procedure.

Definition 3. *A* selection rule *is a function from a multiset of equations S to a nonempty subset T of S, such that if $x \approx t_1 \in T$, $x \in Vars$ and $t_1 \notin Vars$, then either there is another member of T, $x \approx t_2$, or $x \approx t_1$ is in a cycle and t_1 is not boxed. Every equation in T is considered* selected.

Notice that if $x \approx t$ is in a solved form, it cannot be selected.

We will prove that *BSM* always halts on a goal G, and if G is E-unifiable then a normal form will be found.

Definition 4. *A goal G is in* normal form *if the equations of G are all solved and they can be arranged in the form $\{x_1 \approx t_1, \cdots, x_n \approx t_n\}$ such that for all $i \leq j$, x_i is not in t_j.*

Then we define θ_G to be the substitution $[x_1 \mapsto t_1][x_2 \mapsto t_2]\cdots[x_n \mapsto t_n]$. θ_G is a most general E-unifier of G.

One application of any inference rule to the goal G with the resulting goal G', is denoted by $G \to G'$.

5 Completeness and Termination

In this section we prove that if $E = RHS(E')$ where E' is finite and saturated under Paramodulation, then the BSM procedure always terminates in nondeterministic polynomial time, and it finds a normal form if the goal is E-unifiable.

Definition 5. *Let G be a goal and σ be a ground substitution. Then (G, σ) is* reduced *if $x\sigma$ is reduced wrt R_E for all variables $x \in G$, and $t\sigma$ is reduced wrt R_E whenver t is boxed.*

Lemma 2. *Let $E = RHS(E')$, where E' is finite and saturated by Paramodulation. If (G, σ) is reduced, $E \models G\sigma$, G is not in normal form, then there is G' and σ' such that $G \to G'$, (G', σ') is reduced, $E \models G'\sigma'$ and $\sigma' \leq_E \sigma[Var(G)]$.*

Proof. If G is not in a normal form, some equation or equations will be selected and we have several cases to consider. We give the proof of one case, and the others can be found in [13].

1. **$u \approx v$ is selected and u and v are variables**
2. **$u \approx v$ is selected and u, v are not variables**
   Since $E \models u\sigma \approx v\sigma$, there are two possibilities according to Theorem 2:
   (a) **Th.2(1) holds for $u\sigma \approx v\sigma$**
   $u = f(u_1, \cdots, u_n)$, $v = f(v_1, \cdots, v_n)$, and $E \models \bigcup_{i=1}^{n} u_i\sigma \approx v_i\sigma$. Thus $G \to G'$ by Decomposition and $E \models G'\sigma$. (G', σ) is reduced with respect to the variables (we have not changed anything about the variables in this case).
   As for the other terms in G', if $f(u_1, \cdots, u_n)$ was boxed in G,then we assume that $f(u_1, \cdots, u_n)\sigma$ is R_E-reduced. Hence the same can be said about all subterms of $f(u_1, \cdots, u_n)\sigma$. Hence $u_1, \ldots, u_n$, which will be boxed in the result of Decomposition, preserve the property: each $u_i\sigma$, which will be boxed in the conclusion, will also be R_E-reduced.

   (b) **Th.2(2) holds for $u\sigma \approx v\sigma$**
   $u = f(u_1, \ldots, u_n)$ and $v = g(v_1, \ldots, v_m)$, $f(s_1, \ldots, s_n) \approx t \in E$, $E \models \bigcup_{i=1}^{n} u_i\sigma \approx s_i\sigma'$ and $E \models g(v_1, \ldots, v_m)\sigma \approx t\sigma'$, where σ' is an extension of σ for new variables in the terms from E.
   There are two possibilities depending on the form of t:
   i. If $t = g(t_1, \ldots, t_m)$, $E \models \bigcup_{j=1}^{m} v_j\sigma \approx t_j\sigma'$, where $\sigma = \sigma'[Var(G)]$. Hence Mutate is applicable. Either the first case applies, where e.g. $f(u_1, \ldots, u_n)$ is boxed, i.e. $f(u_1, \ldots, u_n)\sigma$ is R_E-reduced, and this

allows to box $f(y_1, \ldots, y_n)$ in the conclusion of the rule, or the second case applies, where both $f(u_1, \ldots, u_n)$ and $g(v_1, \ldots, v_m)$ are unboxed, hence their ground instances are not necessarily R_E-reduced, and $f(y_1, \ldots, y_n)$ cannot be boxed in the conclusion of the rule. Therefore $G \to G'$ and $E \models G'\sigma'$. New terms in G' introduced from E are boxed, because by theorem 2, they are R_E-reduced. Hence (G', σ') is reduced.

ii. If $t = x$, where x is a variable. Then $f(s_1, \ldots, s_n) \approx x \in E$ and $E \models \bigcup_{i=1}^{n} u_i\sigma \approx s_i\sigma'$ and $E \models x\sigma \approx g(v_1, \ldots, v_m)\sigma$. The rule VariableMutate is then applicable, and $G \to G'$ by this rule, $E \models G'\sigma'$. By Theorem 2, all $s_i\sigma$ are R_E-reduced, hence all s_i can be boxed in the conclusion of the rule. (G', σ') is reduced, where $\sigma = \sigma'[Var(G)]$, because of the only new variable x, by theorem 2, we know that $x\sigma$ is R_E-reduced.

3. **$x \approx v$ is selected, where x is a variable, v is not a variable and $x \approx v$ is part of a cycle**
4. **$x \approx v_1$ and $x \approx v_2$ are selected, where x is a variable, and v_1 and v_2 are not variables**

In order to prove that BSM always halts, we define a measure:

Definition 6. *Let M be a measure function from a unification problem G to a triple (m,n,p) of natural numbers, where m is the number of unboxed, non-variable symbols in G, n is the number of non-variable symbols in G, and p is the number of unsolved variables in G.*

Theorem 3. *Let $E = RHS(E')$ where E' is finite and saturated by Paramodulation. Then BSM solves the E-unification problem G in nondeterministic polynomial time.*

Proof. The following table shows how $M(G)$ decreases with the application of each rule, and hence can be compared wrt lexicographic order. For example, Decomposition preserves or decreases the number m of unboxed, non-variable symbols in G, but always decreases the number n of non-variable symbols in G.

	m	n	p
Decomposition	$\geq$	$>$	
Mutate	$>$		
Imitation	$\geq$	$>$	
Mutate&Imitate	$>$		
Variable Elimination	$=$	$=$	$>$
VariableMutate	$>$		
Mutate&Imitate-cycle	$>$		
Imitation-cycle	$>$		

Let a be the greatest arity in the signature of $E \cup G$. To prove the claim, we show that the number, $\mu(G) = (a+2)|E| * m + (a+1)n + p$ is decreased with the application of every rule. Hence the run of the algorithm will take no longer than $O(|G|)$, since a and $|E|$ are constant, and m, n and p are bounded by $|G|$.

In the following, G' is the goal obtained by an application of one inference rule, $G \to G'$, $M(G) = (m, n, p)$, and $M(G') = (m', n', p')$. Missing cases are in [13].

- **Decomposition:** $m' \leq m$, $n' = n - 2$ and $p' \leq p$.
  $\mu(G') \leq (a+2)|E| * m + (a+1)(n-2) + p < (a+2)|E| * m + (a+1)n + p$.
- **Mutate:** $m' \leq m - 1$, $n' \leq n + |E| - 2$, $p' \leq p + |E|$.
  $\mu(G') \leq (a+2)|E| * (m-1) + (a+1)(n + |E| - 2) + p + |E| = (a+2)|E| * m + (a+1)n + p - 2a - 2 < (a+2)|E| * m + (a+1)n + p$.
- **Imitation:** $m' \leq m$, $n' \leq n - 1$, $p' \leq p + a$.
  $\mu(G') \leq (a+2)|E| * m + (a+1)(n-1) + p + a = (a+2)|E| * m + (a+1)n + p - 1 < (a+2)|E| * m + (a+1)n + p$.
- **Mutate&Imitate:** $m' = m - 1$, $n' \leq n + |E| - 1$, $p' \leq p + |E|$.
  $\mu(G') \leq (a+2)|E| * (m-1) + (a+1)(n + |E| - 1) + p + |E| = (a+2)|E| * m + (a+1)n + p - a - 1 < (a+2)|E| * m + (a+1)n + p$.
- **Variable Elimination:** $m' = m$, $n' = n$, $p' = p - 1$.
  $\mu(G') = (a+2)|E| * m + (a+1)n + p - 1 < (a+2)|E| * m + (a+1)n + p$.
- **Imitation-cycle:** $m' = m - 1$, $n' = n$, $p' = p$.
  $\mu(G') = (a+2)|E| * (m-1) + (a+1)n + p = (a+2)|E| * m + (a+1)n + p - (a+2)|E| < (a+2)|E| * m + (a+1)n + p$.

By Lemma 2, the algorithm must halt with a normal form if the goal is E-unifiable, therefore the algorithm runs in nondeterministic polynomial time.

There are several sources of "don't know" non-determinism here:

1. We don't know which equation from E to use for a given form of Mutate rule (Mutate, Mutate&Imitate, VariableMutate or Mutate&Imitate-cycle each taken alone), if several equations are applicable.
2. There may be conflicts between VariableMutate and any of the following Mutate rules: Mutate, Mutate&Imitate, Mutate&Imitate-cycle.
3. Decomposition may be in conflict with Mutate or with VariableMutate.
4. Imitation-cycle may conflict with Mutate&Imitate-cycle or VariableMutate.

6 Achieving determinism

There are 4 sources of non-determinism in the BSM procedure, as explained above. Here further restrictions will be put on E in order to make the algorithm deterministic. The first of these restrictions will address the problem of the choice of equations to use with a form of Mutate, and the second and third will deal with the choice of the inference rules that could be applied to a unification problem.

A set of equations E is *subterm-collapsing* if there are terms t and u such that, t is a proper subterm of u and $t =_E u$.

Definition 7. *We call E* deterministic *if E is not subterm-collapsing and:*

1. *No two equations in E have the same root symbols at their sides. For example, we can't have both $f(a) \approx g(b)$ and $f(c) \approx g(d)$ in E.*

2. *If $s \approx t \in E$, then neither t nor s is a variable*
3. *If $s \approx t \in E$, then $root(s) \neq root(t)$.*

We will show that if $E = RHS(E')$ where E' is saturated under Paramodulation and E is deterministic, then BSM can be turned into a deterministic algorithm, which will mean that the algorithm halts deterministically in a linear number of inference steps. Each step takes no more than linear time, so the algorithm is $O(n^2)$. It will also show that the theory is unitary[1], because we get a most general unifier from the algorithm.

Lemma 3. *Let $E = RHS(E')$, such that E' is finite and saturated by Paramodulation, and E is deterministic. Then in the BSM algorithm for theory E, the rule VariableMutate is not applicable.*

Notice that the elimination of VariableMutate rule removes source 2 and part of source 1 and 3 of non-determinism in the BSM algorithm.

Lemma 4. *Let $E = RHS(E')$, where E' is finite and saturated by Paramodulation, and E is deterministic. Then in the BSM algorithm for the theory E, if Imitation-cycle or Mutate&Imitate-cycle is applicable to a goal G, then G has no solution.*

We define algorithm $BSMd$ the same as algorithm BSM, only without rules VariableMutate, Imitation-cycle and Mutate&Imitate-cycle. Notice that the elimination of cycle-rules completely removes the 4th source and partially the 1st and 2nd source of non-determinism in the BSM algorithm.

Theorem 4. *Let $E = RHS(E')$, where E' is finite and saturated under Paramodulation, and E is deterministic. Then the algorithm $BSMd$ for the theory E solves the E-unification problem G deterministically in $O(|G|)$ inference steps, so in time $O(|G|^2)$. Also, E is unitary.*

Proof. By the completeness argument, the algorithm BSM solves the E-unification problem. By Lemmas 3 and 4, the algorithm $BSMd$ also solves the problem. But there are no sources of non-determinism in the algorithm $BSMd$. Recall the possible sources of non-determinism given at the end of the last section. After the removal of the VariableMutate rule and cycle-rules, we have to consider the following, remaining cases:

1. As for the first source of non-determinism, we are left with a possible conflict of various equations from E used with Mutate or Mutate&Imitate. But Restriction 1 on E in an obvious way rules out these cases. Hence this source of non-determinism disappears. The conflicts with VariableMutate or Mutate&Imitate cycle are no longer there, because the inference rules are no longer there.
2. We got rid of the second source of non-determinism by removing VariableMutate from the $BSMd$ algorithm.

3. As for the third source of non-determinism, we are left with a possible conflict between Decomposition and Mutate. But notice that Restriction 3 on E precludes any such conflict, since Decomposition is used only when an equation in the goal has both sides with the same root symbol.
4. The fourth source of non-determinism disappeared with the removal of the cycle-rules from the $BSMd$ algorithm.

There are no other sources of possible non-determinism in $BSMd$. Hence the algorithm $BSMd$ is deterministic and will only take $O(|G|)$ inference steps. Each step can be done in linear time, so the algorithm is $O(|G|^2)$. Since it is deterministic, it computes a most general unifier.

For subterm-collapsing theories, it is possible to show that all those properties are necessary. For example, [14] exhibits a ground theory that satisfies the second and third properties, but whose unification problem is NP-complete. The theory $E = \{f(x,x) \approx x\}$ satisfies the first and third property, but its unification problem is NP-complete[9]. Also, consider the theory $E' = \{f(x,x) \approx 0\}$. In this case $E = RHS(E') = \{f(x,x) \approx 0, f(x,x) \approx f(x',x')\}$, which satisfies the first two properties, but its unification problem is NP-complete[5]. All of those are subterm-collapsing theories, and we don't know if it is possible to show that a subterm-collapsing theory with the above three properties always has a polynomial time procedure to decide the unification problem. However, we know that it cannot be solved in polynomial time. Consider the theory $E = \{fa \approx a, fb \approx b\}$. This is subterm-collapsing and it satisfies all three properties above, but the goal $fx_1 \approx x_1, \cdots, fx_n \approx x_n$ has a complete set of unifiers of size 2^n.

7 Comparison with Basic Narrowing

We will show some advantages of $BSMd$ over Basic Narrowing, which is defined in Figure 7. The Basic Narrowing rules[6] are presented here in the formalism of constraints. To the right of the bar, we put constraints in the form of substitutions, that composed together give the possible solution. Substitutions from the constraints part are never applied to the goal, hence we prevent any inferences into the substituted terms. This is exactly the same as boxing the terms in $BSMd$. Also, any term into which an inference is made, cannot be a variable, and in $BSMd$ we treat variables as boxed.

As an example, we take $E = \{fa \approx b\}$ and the goal $g(fx_1, \ldots, fx_n) \approx g(fy_1, \ldots, fy_n)\}$.

In this case, $BSMd$ gives us the most general unifier our algorithm in a deterministic way, in polynomial time gives us the most general unifier $[x_1 \mapsto y_1, \ldots, x_n \mapsto y_n]$. The only possible rule to apply is Decomposition.

Basic Narrowing is non-deterministic in this case and will search for the solution in exponential time, applying the Narrowing rule to each fx_i and fy_i. It will find all solutions of the form $\{x_i \mapsto a, y_i \mapsto a \mid i \in N\} \cup \{x_i \mapsto y_i \mid i \in N\}$. for all $N \subseteq \{1, \cdots, n\}$. Therefore, it will find 2^n different unifiers, all of which are subsumed by the one unifier generated by $BSMd$.

Basic Narrowing:

$$\frac{s[u] \approx t, G \mid \tau}{s[x] \approx t, G \mid \tau[x \mapsto r]\sigma}$$

where $l \approx r$ is in E, $l \not\preceq r$, $s \not\preceq t$, $\sigma = mgu_\emptyset(l, u\tau)$ and u is not a variable.

Equality Resolution:

$$\frac{u \approx v, G \,|\, \tau}{G \,|\, \tau\sigma}$$

where $\sigma = mgu_\emptyset(u\tau, v\tau)$.

On the other hand, if we change E to contain $fa \approx fb$ instead of $fa \approx b$, we need to use BSM. There will be exponentially many solutions, but E is not deterministic in this case.

8 Conclusion

This paper gives an algorithm which solves E-unification for a certain class of equational theories in NP, and for a more restricted class of theories in quadratic time. There have been other decidability and complexity results shown for classes of equational theories such as [4, 7, 14, 8, 12]. The classes defined in those other papers are not related to ours, except that [14] shows NP-completeness for theories saturated under Paramodulation.

We have defined an inference system for E-unification called Basic Syntactic Mutation (BSM). We apply BSM to solve E-unification for sets of equations finitely saturated by Paramodulation. BSM resembles the Syntactic Mutation inference rules of [10], but after an inference, the terms introduced by the inference are blocked from further inferences, as in Basic Paramodulation[3, 15]. Therefore, our inference rules will halt on equational theories saturated by Paramodulation in nondeterministic polynomial time, as in [14], giving a decision procedure for E-unification in such theories.

A main interest of our inference system was to find equational theories where E-unification can be solved in polynomial time, and our inference rules were designed with that in mind. We give further restrictions on the equational theory, and we show that with those restrictions, our algorithm will halt in deterministic quadratic time, with a linear number of inference steps, and that such theories are unitary. We call such theories *deterministic*. This means unification in these theories is not much harder than in the empty theory. We conjecture that the complexity of our procedure could be reduced to $O(nlg(n))$ or $O(n)$, as in syntactic unification.

The idea behind our reults on deterministic theories is to deal with equational theories which express non-recursive definitions. For example, the definition of adding elements to a list looks like this:

$$add(x, cons(y, z)) = cons(x, cons(y, z))$$

This theory is deterministic, as would be similar theories consisting of adds and inserts. Many natural theories contain axioms such as these. They may contain other axioms, which destroy the deterministic property, however they may still meet many of the conditions for a deterministic theory. Therefore, it is still possible to use the results in this paper to analyze the determinism in the BSM E-unification algorithm and understand how efficient the algorithm will be.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge, 1998.
2. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. In *Journal of Logic and Computation* 4(3), 1-31, 1994.
3. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic Paramodulation. *Information and Computation Vol. 121, No. 2* (1995) pp. 172–192.
4. H. Comon, M. Haberstrau and J.-P. Jouannaud. Syntacticness, Cycle-Syntacticness and shallow theories. In *Information and Computation* 111(1), 154-191, 1994.
5. Q. Guo, P. Narendran and D. Wolfram. Unification and Matching modulo Nilpotence. In *Proceedings 13th International Conference on Automated Deduction*, Rutgers University, NJ, 1996.
6. J.-M. Hullot. Canonical forms and unification. In *Proc. 5th Int. Conf. on Automated Deduction*, LNCS, vol. 87, pp. 318–334, Berlin, 1980. Springer-Verlag.
7. F. Jacquemard. Decidable approximations of term rewriting systems. In H. Ganzinger, ed., *Rewriting Techniques and Applications, 7th International Conference, RTA-96*, LNCS, vol. 1103, Springer, 362-376, 1996.
8. F. Jacquemard, Ch. Meyer, Ch. Weidenbach. Unification in Extensions of Shallow Equational Theories. In T. Nipkow, ed., *Rewriting Techniques and Applications, 9th International Conference, RTA-98*, LNCS, vol. 1379, Springer, 76-90, 1998.
9. D. Kapur and P. Narendran. Matching, Unification, and Complexity. In SIGSAM Bulletin, 1987.
10. C. Kirchner. Computing unification algorithms. In *Proceedings of the Fourth Symposium on Logic in Computer Science*, Boston, 200-216, 1990.
11. D. E. Knuth and P. B. Bendix. Simple word problems in universal algebra. In *Computational Problems in Abstract Algebra*, ed. J. Leech, 263-297, Pergamon Press, 1970.
12. S. Limet and P. Réty. E-unification by Means of Tree Tuple Synchronized Grammars. In *Discrete Mathematics and Theoretical Computer Science*, volume 1, pp. 69–98, 1997.
13. C. Lynch and B. Morawska. `http://www.clarkson.edu/~clynch/papers/bsm_full.ps/`, 2002.
14. R. Nieuwenhuis. Basic paramodulation and decidable theories. (Extended abstract), In *Proceedings 11th IEEE Symposium on Logic in Computer Science, LICS'96*, IEEE Computer Society Press, 473-482, 1996.
15. R. Nieuwenhuis and A. Rubio. Basic Superposition is Complete. In *Proc. European Symposium on Programming*, Rennes, France (1992).