Decidability and Complexity of Finitely Closable Linear Equational Theories

Conference Paper · March 2001

DOI: 10.1007/3-540-45744-5_43 · Source: CiteSeer

Barbara Morawska
Technische Universität Dresden

Decidability and Complexity of Finitely Closable Linear Equational Theories

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 define a subclass of the class of linear equational theories, called *finitely closable* linear theories. We consider unification problems with no repeated variables. We show the decidability of this subclass, and give an algorithm in PSPACE. If all function symbols are monadic, then the running time is in NP, and quadratic for unitary monadic finitely closable linear theories.

1 Introduction

The problem of E-unification[1] is an important problem for automated deduction, as well as other areas of computer science, such as formal verification and type inference. Given an equational theory E, an E-unifier of terms s and t is a substitution θ such that $s\theta$ and $t\theta$ are equivalent modulo E. In many applications it is necessary to find a *complete set of E-unifiers* of terms s and t, that is, to find a set of E-unifiers of s and t from which all other E-unifiers can be generated.

Unfortunately, E-unification is undecidable in general. In addition, for some equational theories there is no finite complete set of unifiers. Therefore, if it necessary to determine which classes of equational theories have a decidable algorithm and on which E-unification problems. Furthermore, the complexity of those algorithms should be analyzed.

There has been much work in finding particular equational theories with decidable E-unification problems and analyzing their complexity. There has been less work in identifying classes of equational theories with decidable E-unification problems. However, there has been some recent work in that area, but not all of it analyzes complexity. See [9] for some references.

Recently, we have developed a simple new method of E-unification and proved its soundness and completeness[6] for all equational theories. In [7] we have refined it for linear theories. The method is a generalization of the General Mutation inference rules for Syntactic Theories[2–5]. It is an inference procedure that does not always halt. However, the goal of developing this new method was to use it to find decidable classes of equational theories and analyze their complexity, which is what we do in this paper.

** This work was supported by NSF grant number CCR-9712388 .

We consider *linear theories*, i.e., theories where in each equation no terms have repeated variables, although terms on opposite sides of an equation may share variables. This class of equational theories includes all theories with monadic functions symbols, therefore equations on strings can be represented. We only consider E-unification problems whose set of goal equations contains no repeated variables. This is a restricted E-unification problem, but it contains the word problem, which is undecidable for equations on strings, and it also includes some existential problems.

The particular class we prove decidability of is what we call *finitely closable* theories. To use our algorithm, we must assume we know a finite set of terms, such that we can find a complete set of unifiers for each pair of those terms. If the terms that appear in each complete set of unifiers are already in the set, then we call the set finitely closed. When such a set exists, we have an algorithm to solve the E-unification problems mentioned in the previous paragraph. We show the algorithm is in PSPACE. However, for the case of monadic function symbols it is in NP, and furthermore it is quadratic if each complete set of unifiers mentioned above is unitary.

Of course, we have not mentioned, so far, how to find this finite set. We also show some ways in which such a finite set can be found.

The format of the paper is to give some preliminary definitions, then to present the algorithm which gives our decidability results and prove the complexity results. Finally we give a method for finding the finite set in some cases.

2 Preliminaries

We assume we are given a set of variables and a set of uninterpreted function symbols of various arities. An arity is a non-negative integer. *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*. If t is any object, then $Var(t)$ is the set of all variables in t.

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 $E \models G$ means that $E \models e$ for all e in G. If all the function symbols in E are of arity no greater than one, then E is *monadic*.

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 $Dom(\theta) = \{x \mid x\theta \neq x\}$. The *range* of θ, $Ran(\theta)$ is $\{x\theta \mid x \in Dom(\theta)\}$. A substitution σ is *idempotent* if $\sigma\sigma = \sigma$. In this paper, all substitutions will be considered to be idempotent. 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. Whenever an equation or a set of equations has an E-unifier, it also has an idempotent E-unifier. If θ is an E-unifier of $u \approx v$, we say that θ is *linear* if no variable appears more than twice in $Ran(\theta)$, and if a variable z appears twice in $Ran(\theta)$ then there is an x in u and a y in v such that z appears in $x\theta$ and z appears in $y\theta$. This implies that there are not two different variables x and w in u such that z appears in $x\theta$ and $w\theta$.

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 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 $\theta \leq_E \sigma[Var(G)]$.

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.

We say that a term t (or an equation or a set of equations) has *varity* n if each variable in t appears at most n times. An equation $s \approx t$ is linear if s and t are both of varity 1. Note that the equation $s \approx t$ is then of varity 2, but it might not be of varity 1. A set of equations is *linear* if each equation in the set is linear. For example, the axioms of group theory $(\{f(x,, f(y,z)) \approx f(f(x,y),z), f(w,e) \approx w, f(u,i(u)) \approx e$. are of varity 2.

3 Algorithm

We will be considering linear equational theories E. The goals G we are trying to solve are sets of equations with no repeated variables (varity 1). In this section we will give an E-unification algorithm, and in the next section we will prove the algorithm halts for E-closed sets T, defined below, and give the complexity of the algorithm.

Definition 1. *A set of terms T is called E-closed if it satisfies the following conditions:*

1. *every term in T is of varity 1;*
2. *no member of T is a variable;*
3. *if f is a symbol of arity $n \geq 0$ appearing in E, then $f(x_1 \cdots, x_n) \in T$;*
4. *T contains two new constants c and d, which are not symbols of E.*
5. *if s and t are renamings of terms in T, and $\theta \in CSU_E(s,t)$, then θ is linear, and for all x in $Var(s \approx t)$, whenever $x_i\theta$ is not a variable, there is a renaming ρ such that $x_i\theta\rho \in T$;*
6. *if t' is a nonvariable subterm of t, then there is a renaming ρ such that $t'\rho \in T$.*

In the definition of T we assume that we are able to calculate a complete set of E-unifiers for all pairs of terms in T. Each such T could have an associated table listing the complete set of unifiers for each pair of terms in T. If such a T

exists, we will show that the E-unification problem for all goals G of varity 1 is solvable. But first we will show that if G contains symbols that are not in T, then T and its associated table of complete sets of unifiers can easily be extended to handle such goals. First T is extended so that whenever $u[c]$ is a member of T for some term u, then $u[f(t_1, \cdots, t_n)]$ is added to T for every new symbol f, of arity $n \geq 0$, appearing in G. Then the table of complete sets of E-unifiers is extended as follows.

Let $f(x_1, \cdots, x_n)$ and $g(y_1, \cdots, y_m)$ be terms in the extended T, such that f and g are different symbols, and at least one of f and g did not exist in E. If f is not a symbol in E, then let $u = c$, else let $u = f(x_1, \cdots, x_n)$. If g is not a symbols in E, then let $v = d$, else let $v = f(y_1, \cdots, y_m)$. Find the complete set of E-unifiers $\{\sigma_1, \cdots, \sigma_k\}$ of u and v. Let $\{\theta_1, \cdots, \theta_k\}$ be the set of substitutions such that each θ_i is created from σ_i by replacing each occurrence of c in the range of σ_i by $f(x_1, \cdots, x_n)$, and replacing every occurrence of d in the range of σ_i by $g(y_1, \cdots, y_m)$. Then $\{\theta_1, \cdots, \theta_k\}$ is a complete set of E-unifiers for $f(x_1, \cdots, x_n) \approx g(y_1, \cdots, y_m)$. Furthermore, all terms in the range of each θ_i have already been added to T.

Again, let f be a symbol in G that is not in E. Then, a complete set of E-unifiers for $f(x_1, \cdots, x_n)$ is $\{[x_1 \mapsto z_1, \cdots, x_n \mapsto z_n, y_1 \mapsto z_1, \cdots, y_n \mapsto z_n]\}$. All terms in the range of this substitution are variables.

Now we have an extended T which is E-closed over the symbols of $E \cup G$, and we have an extended table of complete sets of E-unifiers. For the rest of this paper, we will assume we are working with this extended set.

We give several examples of E-closed sets.

Example 1. Let E be the theory of associativity and commutativity, $\{f(f(x, y), z) \approx f(x, f(y, z)), f(x, y) \approx f(y, x)\}$. Let $T = \{f(x, y), c, d\}$. Then any pair of terms where one of them is c or d has no E-unifiers. So, to prove that T is E-closed we only need to check $CSU_E(f(x_1, x_2), f(y_1, y_2))$. In fact, $CSU_E(f(x_1, x_2), f(y_1, y_2)) = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7\}$, where

- $\sigma_1 = [x_1 \mapsto f(z_1, z_2), x_2 \mapsto f(z_3, z_4), y_1 \mapsto f(z_1, z_3), y_2 \mapsto f(z_2, z_4)]$
- $\sigma_2 = [x_1 \mapsto z_2, x_2 \mapsto f(z_3, z_4), y_1 \mapsto z_3, y_2 \mapsto f(z_2, z_4)]$
- $\sigma_3 = [x_1 \mapsto z_1, x_2 \mapsto f(z_3, z_4), y_1 \mapsto f(z_1, z_3), y_2 \mapsto z_4]$
- $\sigma_4 = [x_1 \mapsto f(z_1, z_2), x_2 \mapsto z_4, y_1 \mapsto z_1, y_2 \mapsto f(z_2, z_4)]$
- $\sigma_5 = [x_1 \mapsto f(z_1, z_2), x_2 \mapsto z_3, y_1 \mapsto f(z_1, z_3), y_2 \mapsto z_2]$
- $\sigma_6 = [x_1 \mapsto z_2, x_2 \mapsto z_3, y_1 \mapsto z_3, y_2 \mapsto z_2]$
- $\sigma_7 = [x_1 \mapsto z_1, x_2 \mapsto z_4, y_1 \mapsto z_1, y_2 \mapsto z_4]$

Notice that whenever a nonvariable term appears in the range of some σ_i, then a renaming of that term appears in T. Therefore, T is E-closed.

Example 2. Let E be the monadic theory $\{fgfx \approx gfgx\}$. Let $T = \{fx, gy, fgz, gfw, c, d\}$. Then again, any pair where one term is c or d is not unifiable. The complete set of unifiers of any term with a renaming of itself, such as fx_1 and fx_2, has as most general E-unifier, $[x_1 \mapsto z, y_1 \mapsto z]$. There are twelve more pairs that must be checked. For example $CSU_E(fx, gy) = \{[x \mapsto gfz, y \mapsto fgz]\}$. Also

$CSU_E(fx, gfy) = \{[x \mapsto gfz, y \mapsto gz]\}$. Also $CSU_E(fgx, gfy) = \{[x \mapsto gz, y \mapsto fz]\}$. We leave it to the interested reader to check the others. Notice that any term that appears in the range of a unifier is a renaming of something in T. So T is E-closed.

Example 3. Let E be the monadic theory $\{fggx \approx gffx\}$. Let $T = \{fx, gy, ffz, ggw, c, d\}$. Once again, any pair involving c or d is not E-unifiable. A pair of two renamings of the same term is as in the previous example. The pair of terms fx and gy has a most general E-unifier $[x \mapsto ggz, y \mapsto ffz]$. No other pair of terms is E-unifiable. Therefore, to show that T is E-closed, we only need to verify that a renaming of ggz and ffz are in T.

Example 4. Let $E = \{fx \approx x\}$. Let $T = \{fx, c, d\}$. Then $fx \approx fy$ has a most general E-unifier $[x \mapsto z, y \mapsto z]$. Also, $c \approx fx$ has a most general E-unifier $[x \mapsto c]$. The other complete sets of E-unifiers are easy. T is E-closed, because the only nonvariable terms which appear in the range of a unifier in a complete set of E-unifiers are c and d. Now, suppose we want to consider a goal containing a new monadic function symbol g. First, we add gy to T. Then we note that $c \approx fx$ has a most general E-unifier $[x \mapsto c]$. Therefore, $[x \mapsto gy]$ must be a most general E-unifier of $gy \approx fx$. So the extended set is also E-closed.

We define a function called H to calculate the height of a term in terms of the set T. The height is defined so that a term from T is considered as if it was a single symbol.

Definition 2. *Let T be an E-closed set of terms. $H(t)$ is defined recursively in terms of T.*

1. $H(x) = 0$*, if x is a variable;*
2. $H(s, \rho) = 1 + max\{H(x\rho) \mid x \in Var(s)\}$*, if ρ is a substitution;*
3. $H(t) = min\{H(s, \rho) \mid t = s\rho \text{ and } s \in T\}$ *if there exists an $s \in T$ and a ρ such that $s\rho = t$;*

Note that item 3 applies to a term t if the root symbol of f is in E, since we have said that $f(x_1, \cdots, x_n) \in T$ for all symbols f in E. If T is extended to include symbols in t as explained above, then item 3 always applies. If E is empty, then this definition gives the standard definition of the height of a term, which we denote $SH(t)$. The height of a term is the minimum number of applications of terms in T it takes to construct the term. If $H(t) = n$, we say that the *T-height of t* is n. If $SH(t) = m$, we say the *standard height of t* is m.

Example 5. For example, consider the set T to be $\{fx, gy, fgz, gfw, c, d\}$. Then the T-height $H(x) = 0$ and $H(fx) = H(gx) = H(fgx) = H(gfx) = 1$. The following set of terms are all of T-height 2:
$\{ffx, ggx, ffgx, gfgx, fgfx, gfgx, fgfgx, gffgx, fggfx, gfgfx\}$.

Let $h = max\{SH(t) \mid t \in T\}$. We can see from the definition that $H(t) \leq SH(t)$ and $SH(t) \leq h \times H(t)$.

As for height, we define the standard size of a term and the T-size of a term.

Definition 3. *Let T be an E-closed set. The T-size of a term t, $|t|$, is defined recursively as:*

1. $|x| = 0$, *for any variable x;*
2. $|s|_\rho = 1 + \Sigma\{|x\rho| \mid x \in Var(s)\}$, *if ρ is a substitution;*
3. $|t| = min\{|s|_\rho \mid t = s\rho \text{ and } s \in T\}$ *if there exists an $s \in T$ and a ρ such that $s\rho = t$;*

If $E = \emptyset$, then $|t|$ is the standard size of t. The T-size is related to the standard size in the same way as the T-height is related to the standard height.

If an E-closed set T is finite and G has no repeated variables, then we will prove that we can solve the E-unification problem for G. For the rest of this section, we will assume that T is closed and finite. Since G has no repeated variables, each equation in G can be solved separately without affecting the other results, so for simplicity we will assume that G is a single equation.

An equation $x \approx t$, where x is a variable, is called a *solved* equation.

Our algorithm is based on the following inference rule:

Suppose the goal is $u \approx v$. Let s and t be terms in T, and let ρ be a substitution such that $s\rho = u$ and $t\rho = v$, and such that $H(s, \rho) = H(u)$ and $H(t, \rho) = H(v)$. [1] We don't-know non-deterministically find a unifier $\sigma \in CSU_E(s, t)$. If $Var(s \approx t) = \{x_1, \cdots, x_n\}$ then the rule is the following:

Mutate

$$\frac{u \approx v}{\bigcup_{1 \leq i \leq n} x_i\rho \approx x_i\sigma}$$

Here is an example.

Example 6. Let $E = \{fgfx \approx gfgx\}$ and let T be the E-closed set $\{fx, gy, fgz, gfx, c, d\}$. Suppose that the goal is $fa \approx gb$. Then $CSU_E(fx, gy) = \{\sigma\}$, where $\sigma = [x \mapsto gfz, y \mapsto fgz]$. We also find a matcher $\rho = [x \mapsto a, y \mapsto b]$ such that $fa = fx\rho$ and $gb = gy\rho$. The Mutate inference rule applies:

$$\frac{fa \approx gb}{a \approx gfz, fgz \approx b}$$

This is because of the fact that $x\rho = a$, $x\sigma = gfz$, $y\sigma = gfz$, and $y\rho = b$.

It is obvious from this example that our inference rule is a generalization of the Mutate Rule from [7].

Consider a related example.

Example 7. Let E and T be as in the above example. Suppose that the goal is $fga \approx gfb$. Then $CSU_E(fgx, gfy) = \{\sigma\}$, where $\sigma = [x \mapsto fz, y \mapsto gz]$. We also find a matcher $\rho = [x \mapsto a, y \mapsto b]$ such that $fga = fgx\rho$ and $gfb = gfy\rho$. The Mutate inference rule applies:

[1] This means that we use the same s, t and ρ as in the definition of T-height.

$$\frac{fga \approx gfb}{a \approx fz, gz \approx b}$$

This is because of the fact that $x\rho = a$, $x\sigma = fz$, $y\sigma = gz$, and $y\rho = b$. In this example, if we chose $s = fx$, $t = gy$, and $\rho = [x \mapsto ga, y \mapsto fb]$, then it would have still been true that $s\rho = fga$ and $t\rho = gfb$. However, this would not have minimized the T-height, so it is not valid.

Mutate always applies to a goal $u \approx v$, because of the definition T, as long as T is extended to cover all the symbols that appear in $u \approx v$ but do not appear in E, as explained above.

We also have an inference rule:

Clash

$$\frac{u \approx v \cup G}{\perp}$$

if there is an s and t with $s\rho = u, t\rho = v$, and s and t are not E-unifiable. If the symbol $\perp$ appears in a goal, then that goal will never yield an E-unifier. An example is:

Example 8. Let $E = \{fggx \approx gffx\}$. Let T be the E-closed set $\{fx, gy, ffz, ggw, c, d\}$. Suppose that the goal is $ffa \approx ga$. If $s = ffz$ and $t = gy$. Then $\rho = [z \mapsto a, y \mapsto a]$ is a matcher. But ffz and gy are not unifiable. The Clash rule applies:

$$\frac{ffa \approx ga}{\perp}$$

So ffa and ga are not E-unifiable. Interestingly, we could have chosen fx and gy from T. Those terms are E-unifiable. Therefore Mutate would have applied. If we kept applying the inference rules in that fashion, then we would not halt. That is why it is necessary to choose s and t to minimize the T-height, and why it is necessary that T is closed in order for this algorithm to halt.

We now prove the soundness of our inference rule.

Theorem 1. *Let s, t, u and v be terms, and let ρ, σ and θ be substitutions such that $s\rho = u$, $t\rho = v$, and $\sigma \in CSU_E(s,t)$. Suppose that for all $x \in Var(s \approx t)$, $x\rho\theta =_E x\sigma\theta$. Then $u\theta =_E v\theta$,*

Proof. Since $x\rho\theta =_E x\sigma\theta$ for all variables in s and t, then by the properties of substitutions: $s\rho\theta =_E s\sigma\theta$ and $t\rho\theta =_E t\sigma\theta$. Hence $u\theta = s\rho\theta =_E s\sigma\theta =_E t\sigma\theta =_E t\rho\theta = v\theta$. (Here the third equality holds because $\sigma \in CSU_E(s,t)$). □

Now we prove the completeness of the rule.

Theorem 2. *Suppose there exists θ such that, $u\theta =_E v\theta$, and there is a matcher ρ, such that, $s\rho = u$ and $t\rho = v$, for some $s,t \in T$. Then there must be a substitution $\sigma \in CSU_E(s,t)$, such that $x\rho\theta =_E x\sigma\theta$ for all variables in $Var(s,t)$.*

Proof. Since $u\theta =_E v\theta$, and ρ is the matcher, $s\rho\theta = t\rho\theta$. Hence there must be a $\sigma \in CSU_E(s,t)$, such that, $\sigma\tau =_E \rho\theta$, Then $x\rho\theta =_E x\sigma\tau = x\sigma\sigma\tau$, since we assume every substitution is idempotent. Furthermore, $x\sigma\sigma\tau =_E x\sigma\rho\theta = x\sigma\theta$, because ρ does not apply to any variables in $Ran(\sigma)$. $\square$

Our algorithm is defined in terms of the Mutate inference rule:

$$\frac{u \approx v}{\bigcup_{1\leq i\leq n} x_i\rho \approx x_i\sigma}$$

Recall that $u = s\rho$ and $v = t\rho$. Since s and t are from T, and we are assuming that $u \approx v$ has no repeated variables, we can divide the variables $\{x_1, \cdots, x_n\}$ into disjoint sets Y and Z such that Y contains all the variables in s, and Z contains all the variables in t.

Then the algorithm we will describe in this section is as follows. Suppose we want to solve the E-unification problem for a single equation $u \approx v$. If u is a variable, then we return the substitution $[u \mapsto v]$. If v is a variable we return $[v \mapsto u]$. Otherwise, find an s and t as required in the inference rule. Then for every $\sigma \in CSU_E(s,t)$ we will recursively solve $z\sigma \approx z\rho$ for all $z \in Z$. Assume these recursive calls to solve $z\sigma = z\rho$ all return an E-unifier. Then let θ' be the union of all the unifiers. Since $u \approx v$ will be assumed to have no repeated variable, and since each substitution in the complete set of unifiers of two terms in T will be linear, the union is well-defined.[2] Then we apply θ' to each equation $y_j\rho \approx y_j\sigma$. The result of the application of θ' will be $y_j\rho \approx y_j\sigma\theta'$, since θ' does not apply to any of the variables in the range of ρ. Let θ'' be the union of all of these unifiers obtained from recursive calls on $y_j\rho \approx y_j\sigma\theta'$. Then the unifier of $u \approx v$ is $\theta'\theta''$. If any of the recursive calls returns $\perp$, then *solve* will also return $\perp$. See the algorithm in Figure 1. We must prove that the algorithm will halt. We will prove it halts by giving a bound on the number of recursive calls. In order to do so, we also give a bound on the T-heights of the terms in the ranges of the E-unifiers which are generated.

We make the algorithm nondeterministic by using a choose function.[3] This makes it easier to define. We must take this into account when we analyze the complexity. The function *choose* will select one E-unifier out of a set of E-unifiers. The end of the algorithm results in one E-unifier. Each possible choice in this algorithm would supply a complete set of E-unifiers. This set of E-unifiers may contain some occurrences of $\perp$, since some choices may not give an E-unifier. Then just remove $\perp$ from the set.

In Figure 2, we give an example of performing the algorithm on the goal $fffu_1 \approx gggu_2$, with the equational theory $E = \{fgfx \approx gfgx\}$ and $T = \{fx, gy, fgz, gfw\}$. In this example, after the inference rule, the right branch is always calculated first. That determines a unifier, which is applied to the left branch. Therefore, each left child is shown with the calculated unifier already applied.

[2] The union of anything with $\perp$ is $\perp$.

[3] In an actual algorithm, choose would be replaced by a loop.

```
function solve(u ≈ v)

if u is a variable
    return [u ↦ v]
if v is a variable
    return [v ↦ u]
find s and t, σ and ρ as in definition of inference rule
if s and t are not unifiable
    return ⊥
choose θ' in CSU_E(s ≈ t)
for i = 1 to q
    θ_i = solve(z_i σ ≈ z_i ρ)
θ' = θ_1 ∪ ··· ∪ θ_q
for j = 1 to r
    θ_j = solve(y_j σ ≈ y_j ρθ')
θ'' = θ_1 ∪ ··· ∪ θ_r
return θ'θ''
```

Fig. 1. Algorithm

4 Decidability and Complexity

We will prove that the size of the proof for $u \approx v$ is bounded. The proof is defined as a tree of equations, with $u \approx v$ at the root and for each node e, the children of e are obtained by our inference rule. As we explained in the algorithm, Mutate is applied as long as possible in a depth-first fashion, until we reach leaves of the form $x \approx t$ or $t \approx x$, where x is a variable and t is any term. This defines the mgu θ_i which is applied to the rest of the equations in the goal. The leaves are then counted as solved. Then another equation is selected and the process is repeated. The size of a proof is defined to be the number of non-leaf equations in the proof tree. We will show that if all non-constant function symbols are monadic, then the size of a proof tree of $u \approx v$ is less than or equal to $|u| \times |v|$.

Theorem 3. *Assume that E is a linear equational theory, containing only monadic function symbols, and that T is a finite E-closed set. The size of the proof-tree of a goal of varity 1, $u \approx v$, is less than or equal to $|u| \times |v|$. If x and y are variables in u and v respectively, and θ is a unifier of u and v obtained in the proof, then $|x\theta| \leq |v|$ and $|y\theta| \leq |u|$.*

Proof. The proof will be by induction on the sum of sizes of the terms in the equation $u \approx v$, i.e., $|u| + |v|$. The base case is when $|u| = 0$ or $|v| = 0$. In that

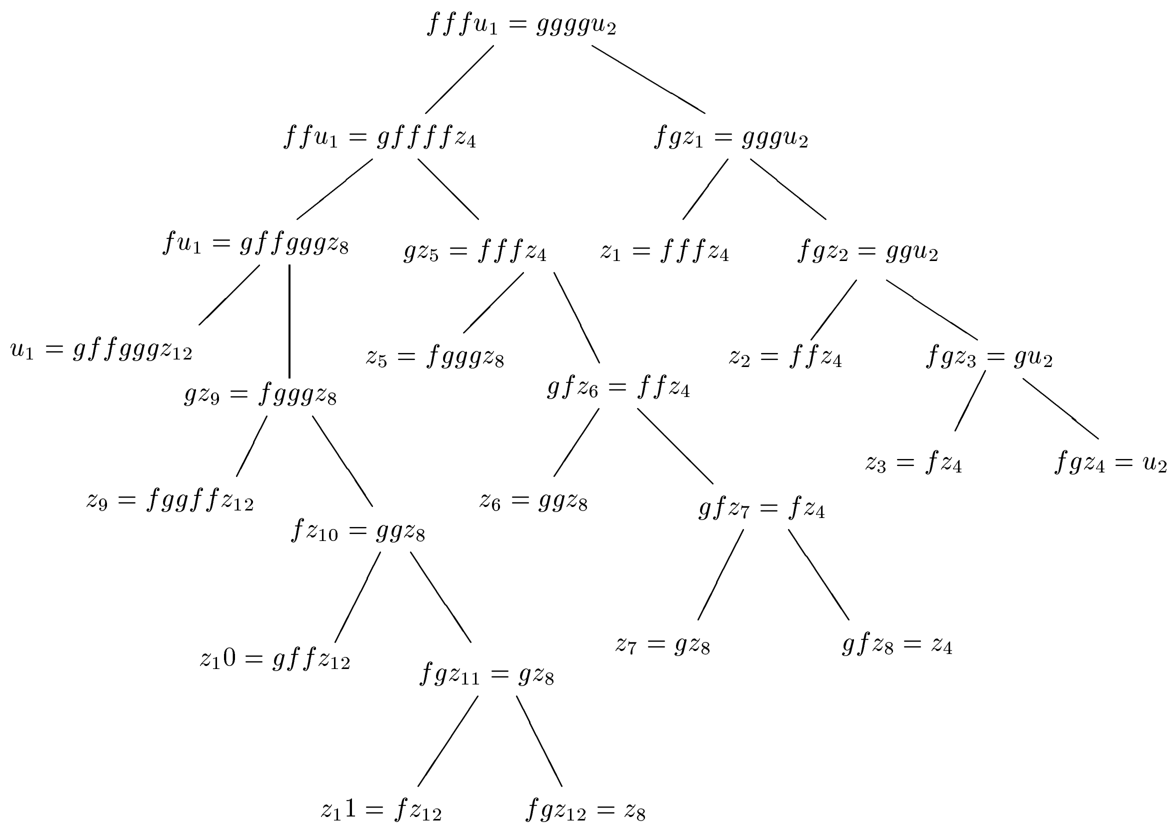

Fig. 2. Proof Tree

case, $u \approx v$ is in normal form.[4] Therefore, the proof is of size 0, since we ignore leaf nodes in the tree.

Now assume that $|u| > 0$ and $|v| > 0$. Assume that the theorem is true for each equation with sum of term sizes smaller than $|u| + |v|$. First we must prove that induction is applicable, i.e. that the size is decreased with the application of the inference rule. An application of the rule with monadic terms will have the following form:

[4] Since $u \approx v$ has no repeated variables, it cannot be of the form $x \approx w[x]$ for some term w.

$$
\begin{array}{ll}
\multicolumn{2}{c}{s\rho[x_u] \approx t\rho[y_v]} \\
\hline
x_s\rho[x_u] \approx x_s\sigma[z_1] & \quad y_t\sigma[z_1] \approx y_t\rho[y_v] \\
 & \qquad\qquad \vdots \\
 & \qquad\qquad \theta_1 \\
x_s\rho[x_u] \approx x_s\sigma[z_1]\theta_1 & : \text{new goal-equation} \\
\qquad\quad \vdots & \\
\qquad\quad \theta_2 &
\end{array}
$$

where $u \approx v$ is our goal, $s, t \in T$, $s\rho = u$, $t\rho = v$, x_u is the only variable in u, y_v is the only variable in v, x_s is the only variable in s, y_t is the only variable in t, z_1 is a variable possibly introduced by the unifier σ of $s[x_s]$ and $t[y_t]$.[5]

In order to apply induction, we need to establish that the size of an equation gets smaller with the application of the rule.

Claim. $|y_t\sigma| + |y_t\rho| < |s\rho| + |t\rho|$.

Proof of Claim. $|y_t\sigma| \leq 1$, because $y_t\sigma \in T$ or y_t is a variable. $|y_t\rho| = |t\rho| \perp 1$, because by definition: $|t\rho| = 1 + |y_t\rho|$. Hence $|y_t\sigma| + |y_t\rho| \leq 1 + |t\rho| \perp 1 = |t\rho| < |t\rho| \leq |s\rho| + |t\rho|$, because $s\rho \geq 1$, since s is not a variable. □

Having proved this lemma, we can state, by the induction assumption, that the size of the proof-tree for $y_t\sigma \approx y_t\rho$ is less than or equal to $|y_t\sigma| \times |y_t\rho| \leq 1 \times (|t\rho| \perp 1) = |t\rho| \perp 1$. Also, $|z_1\theta_1| \leq |y_t\rho| = |t\rho| \perp 1$ and $|y_v\theta_1| \leq |y_t\sigma| \leq 1$, where θ_1 is the unifier obtained in the proof.

Claim. $|x_s\rho| + |x_s\sigma\theta_1| < |s\rho| + |t\rho|$.

Proof of Claim. By the definition of the size of term: $|x_s\rho| = |s\rho| \perp 1$. (This is because: $|s\rho| = 1 + |x_s\rho|$, where s is in T.) The size of the term: $|x_s\sigma[z_1]\theta_1| = 1 + |z_1\theta_1|$, because $x_s\sigma \in T$ or is a variable. We have showed that $|z_1\theta_1| \leq |t\rho| \perp 1$. Hence, $|x_s\sigma\theta_1| \leq 1 + |t\rho| \perp 1 = |t\rho|$. Taking together the sizes of these two terms, we get: $|x_s\rho| + |x_s\sigma\theta_1| \leq |s\rho| + |t\rho| \perp 1 < |s\rho| + |t\rho|$. □

If follows from this claim that the size of the proof tree for $x_s\rho \approx x_s\sigma\theta_1$ is less than or equal to $|x_s\rho| \times |x_s\sigma\theta_1| = (|s\rho| \perp 1) \times |t\rho|$. Also, $|x_u\theta_2| \leq |x_s\sigma\theta_1| \leq |t\rho|$ and $|z_2\theta_1| \leq |x_s\rho| = |s\rho| \perp 1$, where z_2 is a variable possibly introduced by the substitution θ_1.

Taking together these two statements, we can assess the size of the proof-tree for $s\rho \approx t\rho$. It is less than or equal to $1 + |t\rho| \perp 1 + ((|s\rho| \perp 1) \times |t\rho|) = |s\rho| \times |t\rho|$. Also, $|x_u\theta_1\theta_2| = |x_u\theta_2| \leq |t\rho|$ and $|y_v\theta_1\theta_2| = |y_v\theta_1[z_2]\theta_2| \leq |y_v\theta_1| + |z_2\theta_2| \leq 1 + |s\rho| \perp 1 = |s\rho|$. □

The theorem gives us the first major complexity results of the paper.

[5] Technically, we need to show that the new equations generated are of varity 1. We show this in the full paper[8].

Theorem 4. *Let $u \approx v$ be a goal with no repeated variables. Let E be a linear equational theory, containing only monadic function symbols. Let n be the size of $u \approx v$, defined in the standard way. Then*

- *The nondeterministic algorithm in Figure 1 finds a set of E-unifiers for $u \approx v$ in nondeterministic time $O(n^2)$.*
- *Any E-unifier that is constructed is of size $O(n)$.*
- *If every pair of terms in T has a most general E-unifier, then the algorithm is deterministic, and runs in deterministic time $O(n^2)$.*

In order to deal with the more general case of non-monadic terms, we will be considering height of a term and height of a proof-tree, in order to get an idea about the complexity of the procedure. The height of a term was defined earlier. The height of a proof tree is,the length of the longest branch in the proof-tree, excluding its leaf. We write the height of the proof-tree of $u \approx v$ as $H(u \approx v)$.

The general case of the application of our rule is as in the following diagram:

$$\begin{array}{ll}
\multicolumn{2}{c}{s\rho \approx t\rho} \\ \hline
\bigcup_{i=1}^{q} x_i^s\rho \approx x_i^s\sigma & \bigcup_{i=1}^{r} y_i^t\sigma \approx y_i^t\rho \\
 & \qquad\quad \vdots \\
 & \qquad\quad \theta_1 \\
\bigcup_{i=1}^{q} x_i^s\rho \approx x_i^s\sigma\theta_1 & : \text{new goal-equations} \\
\qquad\quad \vdots & \\
\qquad\quad \theta_2 &
\end{array}$$

where $u \approx v$ is our goal, $s, t \in T$, $s\rho = u$, $t\rho = v$, $x_1^u, \cdots, x_m^u$ are the variables in u, $y_1^v, \cdots, y_n^v$ are the variables in v, $x_1^s, \cdots, x_q^s$ are the variables in s, $y_1^t, \cdots, y_r^t$ are the variables in t, $z_1, \cdots z_p$ are variables possibly introduced by the unifier σ of s and t.

Theorem 5. *Assume T is a finite E-closed set. E is linear, and the goal $u \approx v$ is of varity 1, where u and v are not both variables. The height of a proof-tree of $u \approx v$ is less than or equal to $H(u) + H(v) \perp 1$. If $x_1^u, \cdots x_m^u$ and $y_1^v, \cdots, y_n^v$ are variables in u and v respectively, and θ is a unifier of u and v obtained in the proof, then $H(x_i^u\theta) \leq H(v)$ and $H(y_j^v\theta) \leq H(u)$.*

Proof. The proof will be by induction on $H(u) + H(v)$. The base case is when $H(u) = 0$ or $H(v) = 0$. In that case $u \approx v$ is in normal form. Therefore the proof is of height 0, since we ignore leaf nodes when calculating height.

Now assume that $H(u) > 0$ and $H(v) > 0$. Assume that the theorem is true for each equation with sum of heights smaller then $H(u) + H(v)$. First let us consider the right equation: $y_i^t\sigma \approx y_i^t\rho$.

Claim. $H(y_i^t\sigma) + H(y_i^t\rho) < H(s\rho) + H(t\rho)$

Proof of Claim. $H(y_i^t\sigma) \leq 1$, because $y_i^t\sigma$ is in T or is a variable. $H(y_i^t\rho) \leq H(t\rho) \perp 1$, because, according to the definition of height, $H(t\rho) = 1 + max\{H(y_i^t\rho)\}$. Hence $H(y_i^t\sigma) + H(y_i^t\rho) \leq 1 + H(t\rho) \perp 1 = H(t\rho) < H(s\rho) + H(t\rho)$. □

By the induction assumption, if $H(y_i^t\sigma \approx y_i^t\rho) \neq 0$, then $H(y_i^t\sigma \approx y_i^t\rho) \leq H(y_i^t\sigma) + H(y_i^t\rho) \perp 1$, for every $i \in \{1, \cdots, r\}$. We know $H(y_i^t\sigma) \leq 1$, and we know $H(y_i^t\rho) \leq H(t\rho) \perp 1$. Hence, we know that the height of this proof-tree is: $H(y_i^t\sigma \approx y_i^t\rho) \leq 1 + H(t\rho) \perp 1 \perp 1 = H(t\rho) \perp 1$. If $H(y_i^t\sigma \approx y_i^t\rho) = 0$, then $H(y_i^t\sigma \approx y_i^t\rho) = 0 \leq H(t\rho) \perp 1$, since $H(t\rho) \geq 1$.

By induction we also know that:

- $H(z_j\theta_1) \leq H(y_i^t\rho) \leq H(t\rho) \perp 1$ for each z_j in $y_i^t\sigma$,

- $H(y_j^v\theta_1) \leq H(y^t\sigma) \leq 1$ for each y_j^v in $y_i^t\rho$.

Now, consider the left part of the proof-tree.

Claim. $H(x_i^s\rho) + H(x_i^s\sigma\theta_1) < H(s\rho) + H(t\rho)$

Proof of Claim. $H(x_i^s\rho) \leq H(s\rho) \perp 1$, from the definition of height. $H(x_i^s\sigma\theta_1) \leq H(x_i^s\sigma) + max\{H(z_i\theta_1)\}$, where $\{z_1, \cdots, z_k\}$ are the variables in $s\sigma$. $max\{H(z_i\theta_1)\} \leq H(t\rho) \perp 1$, from the analysis of the right equation. Hence $H(x_i^s\sigma\theta_1) \leq 1 + H(t\rho) \perp 1 = H(t\rho)$. Therefore, $H(x_i^s\rho) + H(x_i^s\sigma\theta_1) \leq H(s\rho) \perp 1 + H(t\rho) < H(s\rho) + H(t\rho)$. □

Hence, by the induction assumption, we know that, if $H(x_i^s\rho \approx x_i^s\sigma\theta_1) \neq 0$, then $H(x_i^s\rho \approx x_i^s\sigma\theta_1) \leq H(x_i^s\rho) + H(x_i^s\sigma\theta_1) \perp 1$. Now, $H(x_i^s\rho) \leq H(s\rho) \perp 1$, because according to the definition of height of a term, $H(s\rho) = 1 + max\{x_i^s\rho\}$. Also, $H(x_i^s\sigma[z_1, \cdots, z_p]\theta_1) \leq 1 + max\{H(z_i\theta_1)\} = H(t\rho)$, because $H(z_i\theta_1) \leq H(t\rho) \perp 1$, by the previous lemma. Hence, the height of this proof-tree will be:

- $H(x_i^s\rho \approx x_i^s\sigma\theta_1) \leq H(s\rho) \perp 1 + H(t\rho) \perp 1 = H(s\rho) + H(t\rho) \perp 2$.

If $H(x_i^s\rho \approx x_i^s\sigma\theta_1) = 0$ then $H(x_i^s\rho \approx x_i^s\sigma\theta_1) \leq H(s\rho) + H(t\rho) \perp 2$, because $H(s\rho) \geq 1$ and $H(t\rho \geq 1)$.

The induction assumption also states that

- $H(x_j^u\theta_2) \leq H(x_i^s\sigma\theta_1) \leq 1 + max\{z_i\theta_1\} \leq 1 + H(t\rho) \perp 1 = H(t\rho)$, for each x_j^u in $x^s\rho$, and
- $H(z_j'\theta_2) \leq H(x_i^s\rho) \leq H(s\rho) \perp 1$, for all z_j' in $x_i^s\sigma\theta_1$.

We can now prove the main claim:

The height of the proof-tree for $u \approx v$, i.e. for $s\rho \approx t\rho$, is then: $H(s\rho \approx t\rho) \leq 1 + max\{H(x_i^s\rho \approx x_i^s\sigma\theta_1), H(y_i^t\sigma \approx y_i^t\rho)\} \leq 1 + max\{(H(s\rho) + H(t\rho) \perp 2), (H(t\rho) \perp 1)\} = 1 + H(s\rho) + H(t\rho) \perp 2 = H(s\rho) + H(t\rho) \perp 1$. This is because $H(s\rho) + H(t\rho) \perp 2 \geq H(t\rho) \perp 1$, because we assumed $H(s\rho) > 0$.

We only need to prove the claims about the heights of terms:

$H(x_j^u\theta_1\theta_2) = H(x_j^u\theta_2)$, because x_j^u cannot be in the domain of θ_1. By the assumption, $H(x_j^u\theta_2) \leq H(t\rho)$.

$H(y_j^v\theta_1\theta_2) \leq H(y_j^v\theta_1[z_1', \cdots, z_k']) + max\{H(z_j'\theta_2)\} \leq 1 + H(s\rho) \perp 1 = H(s\rho)$. □

This gives us the following complexity result.

Theorem 6. *Let $u \approx v$ be a goal with no repeated variables. Let E be a linear equational theory. Let n be the size of $u \approx v$, defined in the standard way. Then*

- *The nondeterministic algorithm in Figure 1 finds a set of E-unifiers for $u \approx v$ in PSPACE.*
- *The terms in the range of the E-unifier that is constructed are of height $O(n)$.*

5 Finding a Closed Set

We have shown that once you have an E-closed set, then unification problems of varity 1 are solvable, and we have given the complexity of the decision problem in several cases. That all assumes that we know of an E-closed set. That could be the case for some equational theories. But if we don't know whether there is an E-closed set, then in this section we give a method to produce one which will work for some equational theories.

First we show how to construct an E-closed set in an incremental way:

Let T_0 contain all terms of the form $f(x_1, \cdots, x_n)$, where f is a function symbol of arity $n \geq 0$ appearing in E, and $x_1, \cdots, x_n$ are fresh variables. Also, T_0 will contain two fresh constants c and d.

For $i \geq 0$, T_{i+1} is defined as the set of terms such that $t \in T_{i+1}$ if and only if t is a nonvariable such that there exists some u and v in T_i, a variable x appearing in u and a $\sigma \in CSU_E(u \approx v)$ such that t is a renaming of a subterm of $x\sigma$.

Let $T = \bigcup_{i \geq 0} T_i$. Then T is an E-closed set if the complete sets of unifiers for pairs of terms in T are linear. Of course, T might not be finite. But if T is finite, then this gives us a decision procedure for solving the E-unification problem when the goal has no repeated variables.

We still have not said how to find a complete set of E-unifiers for a pair of terms. This problem is undecidable in general, but in some cases it is possible to use a complete algorithm to generate the E-unifiers. One possibility is to use the complete procedure for linear equational theories presented in [7]. The inference system in that paper is a generalization of the General Mutate inference rules of [2–5], but it is complete for all linear equational theories. It uses a form of eager variable elimination which makes it more efficient.

The problem with using a complete inference system is that it may not halt when two terms are not E-unifiable. However, we also need to check cases of non-unifiability for our algorithm. But, inference rules, such as the ones in [7] can be extended to detect non-unifiability in some cases where the procedure would normally not halt. The inference rules are goal directed, in the sense, that it begins with the equation which must be E-unified. As in the algorithm in this paper, an inference rule will be applied to the goal yielding one or more subgoals. Also, as in this paper, one or more rules may apply at each point. So the algorithm amounts to the simultaneous construction of one or more proof-trees. In some cases, it happens that every proof tree contains an equation $u \approx v$

that is a descendant of a renaming of an equation $s \approx t$, such that $s\rho = u$ and $t\rho = v$ for some ρ. In such cases, the algorithm will never halt, and therefore the initial equation is not E-unifiable.

6 Conclusion

Historically, much of the field of automated deduction has focused on inference procedures that search for a proof of a theorem, and not as much effort has been applied to finding methods of proving something is false. However, if these methods can be applied to verification problems and other applications, we believe it is necessary to identify classes of problems where automated theorem provers will halt, and to understand the complexity of these classes. This is a goal of our research.

The problems we considered in this paper are E-unification problems, since equational logic is useful for many applications. The procedure we give in this paper is an adaptation of a more general procedure for E-unification. However, on the class of problems we consider in this paper, we were able to show a measure on certain E-unification problems, such that the inference rules always reduce the measure; therefore it will halt and we can analyze how quickly it will halt, in order to examine the complexity.

Specifically, we introduce a subclass of linear equational theories, called *finitely closable*. We consider goals with no repeated variables. We show that this class is solvable in PSPACE in general. For monadic theories, it is in NP. For unitary monadic theories, it is solvable in $O(n^2)$.

We think this class is interesting. We also think this research raises many questions to be explored further. Which equational theories are in this class? What is a good procedure for finding a finite (or recursive) E-closed set? Can our complexity results be made better? How can this class be expanded?

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge, 1998.
2. C. Kirchner. Computing unification algorithms. In *Proceedings of the First Symposium on Logic in Computer Science*, Boston, 200-216, 1990.
3. C. Kirchner and H. Kirchner. *Rewriting, Solving, Proving.* `http://www.loria.fr/~ckirchne/` , 2000.
4. C. Kirchner and F. Klay. Syntactic Theories and Unification. In *LICS 5*, 270-277, 1990.
5. F. Klay. Undecidable Properties in Syntactic Theories. In *RTA 4*,ed. R. V. Book, LNCS vol. 488, 136-149, 1991.
6. C. Lynch and B. Morawska. Goal-directed E-unification. Submitted.
7. C. Lynch and B. Morawska. Approximating E-unification. Submitted.
8. C. Lynch and B. Morawska. `http://www.clarkson.edu/~clynch/papers/linear_full.ps/`, 2001.
9. R. Nieuwenhuis and A. Rubio. Paramodulation-based Theorem Proving. To appear in *Handbook of Automated Reasoning*, 2001.