Unification and Matching modulo Nilpotence

Qing Guo[1], Paliath Narendran[1]*, and D.A. Wolfram[2]

[1] Institute of Programming and Logics, Department of Computer Science, State University of New York at Albany, Albany, NY 12222, U.S.A.
E-mail: {guo, dran}@cs.albany.edu

[2] Department of Computer Science, The Australian National University, Canberra, ACT 0200, Australia
E-mail: David.Wolfram@cs.anu.edu.au

Abstract. We consider equational unification and matching problems where the equational theory contains a nilpotent function, i.e., a function f satisfying $f(x, x) = 0$ where 0 is a constant. Nilpotent matching and unification are shown to be *NP*-complete. In the presence of associativity and commutativity, the problems still remain *NP*-complete. But when 0 is also assumed to be the unity for the function f, the problems are solvable in polynomial time. We also show that the problem remains in *P* even when a homomorphism is added. Second-order matching modulo nilpotence is shown to be undecidable.

Subject area: MECHANISMS: unification

1 Introduction

Equational unification is an important computational problem in automated theorem proving. Its usefulness derives from the ability to 'build in' many proof steps into the pattern matching algorithm, possibly shortening the search for a proof. As a new practical application, we define a class of set constraints and show that problems in this class can be solved in polynomial time by equational unification.

Incorporating equational theories in unification algorithms has been guided by the properties of the functions that we often encounter in mathematical theories. For instance, it was observed by Plotkin in 1972 [21] that the properties of associativity and commutativity can be built into unification algorithms. This resulted in extensive research into associative-commutative (AC) unification and rewriting. Unity ($x+0 = x$) and idempotence ($x+x = x$) were also of interest and there has been work on unification algorithms that build-in these theories. There are many other properties that can be treated similarly. Several equational theories have been considered in the literature [5, 13]. Decidability and complexity results have been obtained for many of them [15, 23].

* Some of the results reported here are in partial fulfillment of Qing Guo's Ph.D. requirements, and will form part of his dissertation. Paliath Narendran was partially supported by NSF grant CCR-9404930.

In this paper we consider *nilpotence*: the simple theory $f(x,x) = 0$ where 0 is a constant. Examples of nilpotent functions are subtraction ($-$), Boolean 'exclusive-or' ($\oplus$), and set difference ($\setminus$). We show that elementary unification and matching modulo this theory are NP-complete. We also consider the case where the function f additionally satisfies associativity and commutativity. To our surprise, we found that the problems are still NP-complete. However, when 0 is also the unity of f, i.e., $f(x,0) = x$, unification and matching problems can be solved in polynomial time. Furthermore, we show that unification and matching problems remain in polynomial time when a homomorphism is added to the theory. This polynomial time algorithm can be used to solve a subcase of the set constraints problem. This gives the first subclass of set constraints which can be solved by using equational unification in polynomial time.

Second-order matching modulo nilpotence is shown to be undecidable. This provides the first example of a theory whose first-order unification problem is decidable and finitary, but whose second-order matching problem is undecidable.

1.1 Basic Definitions

Definitions of some of the basic terminology of rewrite rule theory that we later use are given below. For a more extensive list of definitions and a survey of basic results the reader is referred to [7].

A term rewriting system $\mathcal{R}$ is a finite set of *oriented* equations $\{l_i \rightarrow r_i | i \in \mathsf{N}\}$ where l_i and r_i are terms, and $Var(r_i) \subseteq Var(l_i)$. These oriented equations are commonly called *rewrite rules* or simply *rules*. A term rewriting system is said to be *convergent* (or *canonical*) if and only if it is *noetherian* and *confluent*; in other words, if a set of rules is convergent then the rewriting process is both finitely terminating and uniquely terminating in all cases. A rule $l \rightarrow r$ in a term rewriting system $\mathcal{R}$ is *optimally reducing* if and only if it satisfies the following property: for every substitution θ, if $\theta(s)$ is irreducible for every proper subterm s of l, then $\theta(r)$ is irreducible too. A term rewriting system $\mathcal{R}$ is *optimally reducing* if and only if every rule in it is optimally reducing [19].

A function f is *nilpotent* if it satisfies the identity $f(x,x) = 0$. As mentioned earlier, an example of such a function is the *exclusive-or* function in Boolean algebra. Nilpotent unification is a (first-order) E-unification problem in the presence of the theory $E = \{f(x,x) = 0\}$. This is a syntactic theory [17]: it is finite, and resolvant, which means that if $s =_E t$ holds, then there is an equational derivation of it which uses at most one application of $f(x,x) = 0$ at the top level. It is also a *shallow* theory [6], i.e., the variables in every equation (in the presentation) do not appear below level 1. Furthermore, it is not hard to see that the term rewriting system $\{f(x,x) \rightarrow 0\}$ is convergent and *optimally reducing* [19].

Given an equational theory E, an *elementary term* is a term built using only the function symbols appearing in E and additional constants. Two terms s and t are said to be *unifiable modulo an equational theory* E if and only if there exists a substitution θ such that $\theta(s) =_E \theta(t)$. The *elementary unification problem* modulo an equational theory E is to determine, given a set of pairs $\{(s_1, t_1), \ldots, (s_n, t_n)\}$, whether there exists σ such that $\sigma(s_1) =_E \sigma(t_1), \ldots,$ $\sigma(s_n) =_E \sigma(t_n)$, where the terms ($s_i$'s and t_i's) are elementary terms. This is in contrast to the *general unification problem* modulo E where the input terms may contain uninterpreted function symbols that do not occur in E. The matching problem is a subclass of the unification problem where the $t_i's$ are ground terms.

We shall show that elementary nilpotent unifiability is an *NP*-complete problem, despite the extreme simplicity of the theory. Membership in *NP* is easy to show, especially since E is optimally reducing. *NP*-hardness is shown by reduction from *THREE SATISFIABILITY* [22]. ACN-unification, where the function is also associative and commutative, is shown to be *NP*-hard by reduction from *3-COLORABILITY* [9].

2 Unification modulo nilpotence

The unifiability problem whose complexity we consider is:

Definition 1. The problem of elementary *NILPOTENT UNIFIABILITY* is defined as follows.
Instance: A set S of pairs of elementary terms, i.e., first-order terms over the signature $\{f, 0\}$ along with additional constants.

Question: Is S E-unifiable where $E = \{f(x, x) = 0\}$?

In the general case, the terms whose unifiability is sought could contain uninterpreted function symbols. We shall use the following *NP*-complete problem in the proof of our result.

Definition 2. The *NP*-complete problem of *THREE SATISFIABILITY* [22] (*3SAT*) is defined as follows.

Instance: A set $U = \{y_1, \ldots, y_n\}$ where $n \geq 0$ of propositional variables, and a finite set B of propositional clauses each of which has the form $\{l_i \vee l_j \vee l_k\}$ where each literal l_i, l_j, and l_k is either a variable in U or a negation of a variable in U.

Question: Is there a truth assignment $\tau : U \to \{\mathsf{T}, \mathsf{F}\}$ such that each clause in B has at least one literal which is true under τ ?

Theorem 3. *NILPOTENT UNIFIABILITY is NP-complete.*

Proof. The problem is in *NP* because the term rewriting system consisting of the rewrite rule

$$f(x, x) \to 0$$

is convergent, and the right-hand side of the rule is an irreducible ground term. Thus the system is *optimally reducing* [19] and the unifiability check is in *NP*. (See also Theorem 5.1 in [8].)

To show *NP*-hardness, the *NP*-complete problem *THREE SATISFIABILITY* will be polynomially reduced to *NILPOTENT UNIFIABILITY*. Let $U = \{x_1, \ldots, x_n\}$ and $B = \{c_1, \cdots, c_m\}$ be any instance of *THREE SATISFIABILITY*. We shall construct using the following encodings an instance of *NILPOTENT UNIFIABILITY* which is unifiable if and only if B is satisfiable. The terms we construct are over the signature $\{f, 0, a\}$ where a is a new constant symbol.

The function *enc* is defined as follows.

- $enc(\mathsf{T})$ is 0.
- $enc(\mathsf{F})$ is a.
- $enc(x)$ is x, provided that x is a positive literal.
- $\neg x$ is encoded by the new variable $\bar{x}$, along with the equation $neg(x) = 0$ where $neg(x)$ is the term $f(f(x, 0), f(a, \bar{x}))$.
  We need such an equation for each variable x_i in the formula to ensure that in a solution to the instance of *NILPOTENT UNIFIABILITY*, x is instantiated to 0 and $\bar{x}$ to a.
- $enc(\{l_i \vee l_j \vee l_k\})$ is

$$f(f(0, f(f(0, enc(l_i)), f(enc(l_j), y))), f(f(0, enc(l_k)), z))$$

  where y and z are unique new variables.

The encoding has the required property that $\theta(enc(\{x_i \vee x_j \vee x_k\})) =_N 0$, where θ is a substitution, if and only if at least one of these equations holds: $\theta(enc(x_i)) =_N 0$, $\theta(enc(x_j)) =_N 0$, or $\theta(enc(x_k)) =_N 0$.

The instance of *NILPOTENT UNIFIABILITY* constructed is

$$S = \{(neg(x_1), 0), \ldots, (neg(x_n), 0), (enc(c_1), 0), \ldots, (enc(c_m), 0)\}.$$

which has the property that S is N-unifiable if and only if B is satisfiable. It is easy to see that S can be constructed in a time which is a linear function of the number of symbols in B. □

It is possible to show that the problem is *NP*-complete even when there is only *one* pair in the input. Given a nonempty set S of terms, we define a meta-term $\mathcal{M}(S)$ as follows.

$$\mathcal{M}(\{t\}) = f(t, 0)$$
$$\mathcal{M}(\{t_1, t_2, \cdots, t_n\}) = f(t_1, f(t_1, \mathcal{M}(t_2, \cdots, t_n))).$$

We now have the following result.

Lemma 4. *Given a set $S = \{t_1, t_2, \cdots, t_n\}$ of terms, $\{(\mathcal{M}(S), 0)\}$ is E-unifiable if and only if $\{(t_1, 0), \ldots, (t_n, 0)\}$ is E-unifiable.*

The theory N can easily be seen to be finitary (i.e., complete sets of unifiers are always finite) by Proposition 2 (page 327) from [12].

3 ACN-unification and matching

Let $+$ be a nilpotent function. We consider the case where $+$ additionally satisfies associativity and commutativity. In other words, the equational theory ACN is

$$
\begin{aligned}
(x+y)+z &= x+(y+z) \\
x+y &= y+x \\
x+x &= 0
\end{aligned}
$$

We can prove the following theorem.

Theorem 5. *Elementary ACN-UNIFIABILITY is NP-complete.*

Proof. The problem is clearly in NP.

We reduce the NP-complete problem *GRAPH 3-COLORABILITY* [9] to that of *ACN-UNIFIABILITY.*

Let graph $G = (V, E)$ and color set $C = \{c_1, c_2, c_3\}$ be any instance of *GRAPH 3-COLORABILITY,* where $V = \{v_1, v_2, \dots, v_n\}$ and $n \geq 3$. We construct an instance of *ACN-UNIFIABILITY* as follows.

We treat the node names $v_1, v_2, \dots, v_n$ as variables. For every edge $e_k = \{v_i, v_j\} \in E$, we construct the pair

$$(v_i + v_j + z_k \ , \ c_1 + c_2 + c_3)$$

where $Z = \{z_1, z_2, \dots, z_m\}$ is a set of extra variables and $m = |E|$. Obviously, every $z_k, 1 \leq k \leq m$, only appears in one pair. The construction results in a set of pairs $S = \{(s_1, t_1), \dots, (s_m, t_m)\}$ where $m = |E|$.

Now we show that an instance of *3-COLORABILITY* is solvable iff S is ACN-unifiable.

If the *3-COLORABILITY* instance is solvable, then there is a substitution $\sigma : V \rightarrow C$ such that $\sigma v_i \neq \sigma v_j$ if $e_k = \{v_i, v_j\} \in E$. We extend this substitution to Z by following: for $1 \leq k \leq m, \sigma z_k = c$, where $(v_i + v_j + z_k, c_1 + c_2 + c_3) \in S$ and $\{c\} = \{c_1, c_2, c_3\} - \{\sigma v_i, \sigma v_j\}$

For any pair $(v_i + v_j + z_k, c_1 + c_2 + c_3) \in S$, $\sigma v_i, \sigma v_j, \sigma z_k \in \{c_1, c_2, c_3\}$ and $\sigma v_i \neq \sigma v_j \neq \sigma z_k$, so $\sigma v_i \cup \sigma v_j \cup \sigma z_k = \{c_1, c_2, c_3\}$.

$$
\begin{aligned}
\sigma(v_i + v_j + z_k) &= \sigma v_i + \sigma v_j + \sigma z_k \\
&=_{ACN} c_1 + c_2 + c_3
\end{aligned}
$$

Therefore σ is an ACN unifier for S.

If the set S is ACN-unifiable, then there is a substitution θ for $V \cup Z$ and $\theta s_i =_{ACN} \theta t_i$ for every $(s_i, t_i) \in S$. Note the s_i of each pair consists of three distinct variable occurrences, the t_i of each pair consists of three distinct constant occurrences, furthermore there is no 0 in the t_i. Hence for each pair $(v_i + v_j + z_k, c_1 + c_2 + c_3) \in S$, $\theta v_i, \theta v_j \in \{c_1, c_2, c_3\}$ and $\theta v_i \neq \theta v_j$. For every edge $e_k = \{v_i, v_j\} \in E$, there is a pair in S. Thus every node has been assigned a color different from those of its neighbours. Therefore θ is a color assignment for the graph and the 3-Colorability problem is solvable.

The construction of S can be done linearly to the size of E. □

The above proof also shows that ACN-matching is NP-complete, because for every $(s_i, t_i) \in S$, t_i is a ground term. A subcase of this problem, where for every $(s_i, t_i) \in S$, both s_i and t_i are non-ground terms, can be shown to be in P, by reduction to $ACUN$-unification (discussed below). We leave this proof to the Appendix.

4 ACUN-unification and matching

When 0 is also the unity of $+$, the theory is

$$\begin{aligned} (x+y)+z &= x+(y+z) \\ x+y &= y+x \\ x+0 &= x \\ x+x &= 0 \end{aligned}$$

We call this theory $ACUN$. Note that exclusive-or ($\oplus$) and symmetric set difference[3] satisfy these properties.

First we have a theorem on reorganizing $ACUN$ equations.

Theorem 6.

$$x_1 + \ldots + x_m + a_1 + \ldots + a_k =_{ACUN} x_{m+1} + \ldots + x_n + a_{k+1} + \ldots + a_l$$
$$\textit{iff}$$
$$x_1 + \ldots + x_m + x_{m+1} + \ldots + x_n =_{ACUN} a_1 + \ldots + a_k + a_{k+1} + \ldots + a_l$$

Proof. Trivial. □

By the above theorem, any $ACUN$ unification equation can be reorganized into an equation in which variables appear in one side and constants appear in another side. We will use this fact in our algorithm. This also shows that any elementary $ACUN$-unification problem can be reduced to an $ACUN$-matching problem.

We now give an algorithm for elementary $ACUN$-unification.

1. Input
   - $X = \{x_1, \ldots, x_n\}$ is the set of variables which appear in S.
   - $C = \{c_1, \ldots, c_l\}$ is the set of constants which appear in S.
   - $S = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ is the set of unification equations, where t_i's are constant terms and $s_i = x_{i1} + x_{i2} + \ldots + x_{im_i}$, $x_{ij} \in X$.
2. Algorithm
   (a) For each constant $c_i, 1 \leq i \leq l$
      - Create a set of Boolean variables $X^{c_i} = \{x_1^{c_i}, \ldots, x_n^{c_i}\}$.

[3] If A and B are sets, then $(A \setminus B) \cup (B \setminus A)$ is their symmetric set difference.

– Generate a set of equations S^{c_i} from S.

$$\left[\begin{array}{l} x_{11}^{c_i} + x_{12}^{c_i} + \cdots + x_{1m_1}^{c_i} = \begin{cases} 1 & \text{if } c_i \in t_1 \\ 0 & \text{otherwise} \end{cases} \\ x_{21}^{c_i} + x_{22}^{c_i} + \cdots + x_{2m_2}^{c_i} = \begin{cases} 1 & \text{if } c_i \in t_2 \\ 0 & \text{otherwise} \end{cases} \\ \qquad \cdots \qquad\qquad \cdots \\ x_{k1}^{c_i} + x_{k2}^{c_i} + \cdots + x_{km_k}^{c_i} = \begin{cases} 1 & \text{if } c_i \in t_k \\ 0 & \text{otherwise} \end{cases} \end{array}\right.$$

– Solve S^{c_i} by Gaussian elimination over the Boolean ring.

(b) If every $S^{c_i}, 1 \leq i \leq l$, has solutions then θ is a substitution defined as follows:

$$\begin{aligned} \theta(x_1) &= c_1 \cdot x_1^{c_1} + \cdots + c_l \cdot x_1^{c_l} \\ \theta(x_2) &= c_1 \cdot x_2^{c_1} + \cdots + c_l \cdot x_2^{c_l} \\ &\cdots \\ \theta(x_n) &= c_1 \cdot x_n^{c_1} + \cdots + c_l \cdot x_n^{c_l} \end{aligned}$$

Theorem 7. *$S^{c_1}, S^{c_2}, \ldots, S^{c_l}$ have solutions iff S is ACUN unifiable and $\theta(s_i) =_{ACUN} \theta(t_i)$, $1 \leq i \leq k$. The complexity of above algorithm is $O(nk^2l)$.*

Corollary 8. *Elementary ACUN-UNIFIABILITY can be solved in polynomial time.*

However general unification and matching modulo *ACUN* are *NP*-complete.

Theorem 9. *General ACUN-UNIFIABILITY is NP-complete.*

Proof. Membership in *NP* can be shown using techniques from [4]. We omit the details.

We reduce the *NP*-complete problem $3SAT$ to general *ACUN-UNIFIABILITY*. The reduction is similar to the one used in [16] to show *NP*-completeness of set-matching.

Let a set of clauses $C = \{c_1, c_2, \ldots, c_m\}$ be any instance of $3SAT$, over a set of Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$. Each c_i is of the form $\{l_{i_1}, l_{i_2}, l_{i_3}\}$, $i_j \in \{1, \ldots, n\}$, where each l_{i_j} is either x_{i_j} or its complement. We construct an instance of general *ACUN-UNIFIABILITY* as follows.

For every clause $c_i = \{l_{i_1}, l_{i_2}, l_{i_3}\}$, we introduce a distinct function g_i and define seven terms $t_{i_1}, t_{i_2}, \ldots, t_{i_7}$ as follows: $t_{i_j} = g_i(a_{i_1}, a_{i_2}, a_{i_3})$, where $a_{i_k} \in \{0, 1\}$ and $\{a_{i_1}, a_{i_2}, a_{i_3}\}$ is a truth assignment which make c_i true. There are seven different truth assignments that make c_i true, so there are seven different ground terms $t_{i_1}, t_{i_2}, \ldots, t_{i_7}$. Then we define six non-ground terms $s_{i_1}, s_{i_2}, \ldots, s_{i_6}$ as follows: $s_{i_j} = g_i(y^i_{j_1}, y^i_{j_2}, y^i_{j_3})$, where $y^i_{j_k}$'s are new variables. Finally, we construct a pair

$$(g_i(x_{i_1}, x_{i_2}, x_{i_3}) + s_{i_1} + \ldots + s_{i_6}, \quad t_{i_1} + t_{i_2} + \ldots + t_{i_7})$$

The construction results in a set of pairs S where $|S| = m$. Now we show that the instance of $3SAT$ is solvable iff S is $ACUN$ unifiable.

If the $3SAT$ instance is solvable, then there is a truth assignment $\sigma : X \rightarrow \{0,1\}$ such that every $c_i \in C$ is satisfiable. For every pair $(g_i(x_{i_1}, x_{i_2}, x_{i_3}) + s_{i_1} + \ldots + s_{i_6},\ \ t_{i_1} + t_{i_2} + \ldots + t_{i_7})$, apply the truth assignment σ on the variables $x_{i_1}, x_{i_2}, x_{i_3}$. Because σ make c_i true, there must be a $t_{i_j} \in \{t_{i_1}, t_{i_2}, \ldots, t_{i_7}\}$ such that $t_{i_j} = g_i(\sigma x_{i_1}, \sigma x_{i_2}, \sigma x_{i_3})$. The remaining six terms $t_{i_{j'}}$ can be unified with terms $s_{i_1}, \ldots, s_{i_6}$. Thus σ can be extended to a substitution θ such that

$$\theta(g_i(x_{i_1}, x_{i_2}, x_{i_3}) + s_{i_1} + \ldots + s_{i_6}) =_{ACUN} t_{i_1} + t_{i_2} + \ldots + t_{i_7}$$

If the set S is $ACUN$-unifiable, then there is a substitution θ such that for every pair in S

$$\theta(g_i(x_{i_1}, x_{i_2}, x_{i_3}) + s_{i_1} + \ldots + s_{i_6}) =_{ACUN} t_{i_1} + t_{i_2} + \ldots + t_{i_7}$$

For each clause c_i we can find a $t_{i_j} \in \{t_{i_1}, \ldots, t_{i_7}\}$ where $t_{i_j} = g_i(\theta x_{i_1}, \theta x_{i_2}, \theta x_{i_3})$. Now by restricting θ to X we can get a truth assignment that makes every clause $c_i \in C$ true. □

5 ACUNh-unification

Let h be a homomorphism. The equational theory $ACUNh$ is

$$\begin{aligned} (x+y)+z &= x+(y+z) \\ x+y &= y+x \\ x+0 &= x \\ x+x &= 0 \\ h(x+y) &= h(x)+h(y) \\ h(0) &= 0 \end{aligned}$$

We use $h^k x$ to represent term $\underbrace{h(h(\ldots h}_{k}(x)\ldots))$ $(k \geq 0)$. Obviously, $h^{k_1} + h^{k_2} + \ldots + h^{k_n}$ is a polynomial over $\mathbf{Z}_2[h]$. The $ACUNh$-unification problem can be solved by solving equations over $\mathbf{Z}_2[h]$. (The connection between elementary unification problems of theories with homomorphisms and solving linear equations over algebraic domains was first observed by Nutt [20].)

For example, consider the pair $< h(x)+x, h(h(a))+a >$, where a is a constant. Take $x = x_h(a)$. Equation $(h+1)x_h = (h^2+1)$ is generated, which is solvable on $\mathbf{Z}_2[h]$:

$$\begin{aligned} (h+1)x_h &= (h^2+1) \\ (h+1)x_h &= (h+1)^2 \\ x_h &= (h+1) \end{aligned}$$

Therefore $h(x) + x$ and $h(h(a)) + a$ are $ACUNh$ unifiable and $\theta(x) = h(a) + a$ is an $ACUNh$-unifier.

Similarly, consider the pair $< h(h(x)) + x, h(a) + a >$, where a is a constant. Let $x = x_h(a)$. Equation $(h^2 + 1)x_h = h + 1$ is generated. This equation clearly has no solution on $\mathbf{Z}_2[h]$ and therefore $h(h(x)) + x$ and $h(a) + a$ are not $ACUNh$ unifiable.

We now give an algorithm for $ACUNh$-unification.

1. Input
   - $S = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ is the set of pairs to be unified, where

$$\begin{aligned} s_i &= H_{i1}x_{i1} + H_{i2}x_{i2} + \ldots + H_{im_i}x_{im_i}, \quad H_{ij} \in \mathbf{Z}_2[h] \\ t_i &= H'_{i1}c_1 + H'_{i2}c_2 + \ldots + H'_{il}c_l, \quad H'_{ij} \in \mathbf{Z}_2[h] \end{aligned}$$

   - $C = \{c_1, \ldots, c_l\}$ is the set of constants which appear in S.
   - $X = \{x_1, \ldots, x_n\}$ is the set of variables which appear in S.
2. Algorithm
   (a) For each constant $c_i, 1 \leq i \leq l$,
      - Create a set of variables $X^{c_i} = \{x_1^{c_i}, \ldots, x_n^{c_i}\}$.
      - Generate a set of equations S^{c_i} from S, $x_{pq}^{c_i} \in X^{c_i}$.

$$\left[\begin{array}{l} H_{11}x_{11}^{c_i} + H_{12}x_{12}^{c_i} + \cdots + H_{1m_1}x_{1m_1}^{c_i} = H'_{i1} \\ H_{21}x_{21}^{c_i} + H_{22}x_{22}^{c_i} + \cdots + H_{2m_2}x_{2m_2}^{c_i} = H'_{i2} \\ \qquad \ldots \qquad \ldots \\ H_{k1}x_{k1}^{c_i} + H_{k2}x_{k2}^{c_i} + \cdots + H_{km_k}x_{km_k}^{c_i} = H'_{ik} \end{array} \right.$$

      - Solve S^{c_i} over $\mathbf{Z}_2[h]$.

   (b) If every $S^{c_i}, 1 \leq i \leq l$ has solutions then θ is a substitution defined as follows.

$$\begin{aligned} \theta(x_1) &= x_1^{c_1}c_1 + \cdots + x_1^{c_l}c_l \\ \theta(x_2) &= x_2^{c_1}c_1 + \cdots + x_2^{c_l}c_l \\ &\ldots \\ \theta(x_n) &= x_n^{c_1}c_1 + \cdots + x_n^{c_l}c_l \end{aligned}$$

Theorem 10. *$S^{c_1}, S^{c_2}, \ldots, S^{c_l}$ have solutions iff S is $ACUNh$-unifiable and $\theta(s_i) =_{ACUNh} \theta(t_i)$, $1 \leq i \leq k$.*

Since solvability of linear equations over $\mathbf{Z}_2[h]$ is in P [14], we get

Corollary 11. *Elementary $ACUNh$-unifiability problem can be solved in polynomial time.*

General $ACUNh$-unifiability is again NP-complete. The above method of solving the $ACUNh$-unification problem can be generalized to allow more than one homomorphism. However, this involves solving right-linear equations over $\mathbf{Z}_2 < h_1, h_2, \ldots, h_k >$. It is not known whether this problem can be solved in P.

5.1 Application to set constraints

Set constraints have been considered as a simple, accurate and intuitive formalism in program analysis and type inference [1, 11]. Generally, solving a system of set constraints is very expensive. The satisfiability problem of set constraints is $NEXPTIME$-complete in general. Even for systems of set constraints that only contain nullary set constructors (constants) over set operations, the satisfiability problem turns out to be NP-complete [2]. Therefore identifying tractable subclasses of the set constraints problem is important. We define a subcase of the set constraints problem which can be solved using our algorithm for $ACUNh$-unification. Because only finite terms are considered in the unification problem, we restrict the solutions to finite sets.[4]

Let X be a set of variables, Σ a set constructors consisting of nullary set constructors and *one* unary set constructor, and $H = \{\oplus, \emptyset\}$ be the set operations ($\oplus$ is the symmetric set difference operator). Then a subclass $\mathbf{M}$ of set expressions can be defined on X, Σ and H in the usual way. An *equality constraint* over $\mathbf{M}$ is an equation of the form $E = F$ where E, F are set expressions over $\mathbf{M}$. It is obvious that the operator $\oplus$ has the property of $ACUN$. If we interpret $\emptyset$ as 0, nullary set constructors as constants and unary set constructors as homomorphisms then it is easy to see that the satisfiability problem of a system of equality constraints over $\mathbf{M}$ can be reduced to an $ACUNh$-unification problem.

Theorem 12. *The finite satisfiability problem of a system of equality constraints over* $\mathbf{M}$ *can be solved in polynomial time.*

6 Second-order matching modulo nilpotence

6.1 Undecidability of Second-Order Unification

Goldfarb [10] showed that the unification of second-order terms is recursively undecidable by a reduction from Hilbert's Tenth Problem, the general solution of a finite set of Diophantine equations. Any instance of this problem can be written as a finite set of equations of the forms $x_i \cdot x_j = x_k$, $x_i + x_j = x_k$, and $x_i = C_j$ where the x are numerical variables, and the C_j are numerical constants.

Terms are constructed in the proof to represent numerical constants, and disagreement pairs are constructed to represent equations in any instance of Hilbert's Tenth Problem. The encoding used below is a reformulation [24] in simply-typed λ-calculus of Goldfarb's [10]. If C denotes a natural number $c \geq 0$, it is represented by the term $\overline{c}A$, where $\overline{c}$ is the curried typed Church numeral

$$(\lambda ty.\, \underbrace{t(\cdots t}_{c}(y)\cdots)\lambda x.G(A, x))$$

and $\tau(A) = \tau(x) = \tau(y) = \iota$, $\tau(G) = (\iota, \iota \to \iota)$, and $\tau(\overline{c}) = \tau(t) = (\iota \to \iota)$. For example, $\overline{0}A = A$ and $\overline{2}A = G(A, G(A, A))$.

[4] Infinite solutions can be obtained by solving the linear equations of the last section over $Z_2[[x]]$, the ring of power series with coefficients from Z_2 and indeterminate x.

If a finite set of Diophantine equations has r numerical variables, then the representation of the equations has r disagreement pairs of the form $\langle \overline{1} f_i(A), f_i(\overline{1}A)\rangle$ where $\tau(f_i) = (\iota \to \iota)$ and $1 \le i \le r$.

An equation $x_i = C_j$ where x_i is a numerical variable is represented by the pair of terms $\langle f_i(A), \overline{c_j}A\rangle$.

An equation $x_i + x_j = x_k$ is represented by $\langle f_i(f_j(A)), f_k(A)\rangle$. The substitution

$$\{\langle f_i,\ \lambda w_1.\overline{n}w_1\rangle, \langle f_j,\ \lambda w_1.\overline{m}w_1\rangle, \langle f_k,\ \lambda w_1.\overline{p}w_1\rangle\}$$

where $\tau(w_1) = \iota$ is a unifier for the pair of terms if and only if $p = m + n$.

Similarly, an equation $x_i \cdot x_j = x_k$ is represented by the set with disagreement pairs

$$\begin{array}{l}\langle h_l(A, B, G(G(f_k(A), f_j(B)), A)), G(G(A, B), h_l(f_i(A), G(A, B), A))\rangle,\\ \langle h_l(B, A, G(G(f_k(B), f_j(A)), A)), G(G(B, A), h_l(f_i(B), G(A, A), A))\rangle\end{array}$$

where $\tau(h_l) = (\iota, \iota, \iota \to \iota)$ and $l = 2^i 3^j 5^k$. The substitution

$$\{\langle f_i,\ \lambda w_1.\overline{m}w_1\rangle, \langle f_j,\ \lambda w_1.\overline{n}w_1\rangle, \langle f_k,\ \lambda w_1.\overline{p}w_1\rangle, \langle h_l,\ \lambda w_1 w_2 w_3.u\rangle\}$$

where $u = w_3$ if $n = 0$, and if $n > 0$ then u is

$$G(G(w_1, w_2), G(G(\overline{m.1}w_1, \overline{1}w_2), .., G(G(\overline{m.(n-1)}w_1, \overline{n-1}w_2), w_3)\cdots))$$

and $\tau(w_2) = \tau(w_3) = \iota$, is a unifier for the disagreement set if and only if $p = m \cdot n$.

6.2 Undecidability of Second-Order Nilpotent Matching

Let $E = \{F(x, x) = A\}$ be a nilpotence axiom where F is a constant symbol and $\tau(F) = (\iota, \iota \to \iota)$. Suppose that $S = \{\langle s_1,\ t_1\rangle, \ldots, \langle s_n,\ t_n\rangle\}$ is a finite set of pairs of terms over a signature that includes F and A where for all $i : 1 \le i \le n$, $\tau(s_i) = \tau(t_i)$ and the t_i contain no free variables. An instance of *SECOND--ORDER NILPOTENT MATCHING* is to determine for such a set whether there exists a substitution θ such that $\theta s_i =_E t_i$ for every $i : 1 \le i \le n$.

Theorem 13. *SECOND-ORDER NILPOTENT MATCHING is undecidable.*

Proof. Suppose that the disagreement set $\{\langle s_1,\ t_1\rangle, \ldots, \langle s_n,\ t_n\rangle\}$ is a second--order unification problem produced with Goldfarb's encoding of a system of Diophantine equations. Form the disagreement set

$$\{\langle F(s_1, t_1),\ A\rangle, \ldots, \langle F(s_n, t_n),\ A\rangle\}$$

This is an instance of *SECOND-ORDER NILPOTENT MATCHING.*

If a substitution θ is a unifier of $\{\langle s_1,\ t_1\rangle, \ldots, \langle s_n,\ t_n\rangle\}$ then it is a solution of the above second-order nilpotent matching problem.

We now consider the converse. Suppose that a substitution ρ is a solution to the second-order matching problem. We can assume without loss of generality that there are no free variables in $s_i\rho$ and $t_i\rho$ where $1 \le i \le n$.

We recall that the term decomposition transformation [18] replaces an equation of the form $f(u_1, \ldots, u_m) = f(v_1, \ldots, v_m)$ where f is a m-place constant symbol by the equations $u_i = v_i$ where $1 \leq i \leq m$. Repeatedly apply the decomposition transformation to every equation of the form $\rho s_j = \rho t_j$ where $1 \leq j \leq n$, but do not apply it to equations of the form $F(r_1, r_2) = F(r_3, r_4)$. This can be done because the terms ρs_j and ρt_j are first-order terms, and applications of the decomposition transformation must terminate.

As ρ is a solution, we have $\rho s_j =_E \rho t_j$. This implies that if there is an equation formed through term decomposition whose left and right sides have different principal constant symbols, the equation can only have the form $F(r, r) = A$, or $A = F(r, r)$, where r is a term. All subterms which contain F in the set of equations were introduced by components of ρ in which F occurs. If all subterms of the form $F(w_1, w_2)$ in the range of ρ are replaced by A, then the substitution ρ' so formed is also a solution to the second-order matching problem.

To see this, an equation of the form $F(r_1, r_2) = F(r_3, r_4)$ becomes $A = A$ in the term decomposition of some equation $\rho' s_j = \rho' t_j$. An equation $F(r, r) = A$ also becomes $A = A$ in the term decomposition of some equation $\rho' s_j = \rho' t_j$. The same holds for $A = F(r, r)$.

The substitution ρ' is a solution to the second-order matching problem, and also a unifier of the second-order unification problem because $\rho' s_i$ is syntactically identical to $\rho' t_i$ where $1 \leq i \leq n$.

By Goldfarb's Theorem [10] and this reduction, *SECOND-ORDER NILPOTENT MATCHING* is undecidable. □

7 Conclusions

The following table summarizes the complexity results we have obtained so far. In the table, "elementary problem" refers either to an elementary E-unification or to an elementary E-matching problem, and "general problem" refers either to a general E-unification or to a general E-matching problem.

	complexity	
equational theory E	elementary problem	general problem
N	NP-complete	NP-complete
ACN	NP-complete	NP-complete
$ACUN$	P	NP-complete
$ACUNh$	P	NP-complete

We have also shown, in addition, that second-order matching modulo nilpotence (N) is undecidable. The complexity of $ACUN$ with many homomorphisms is open.

We are currently working on an implementation of the algorithm for $ACUNh$-unification, as part of a unification workbench we are developing at the University at Albany (SUNY).

References

1. A. Aiken. Set constraints: results, applications and future directions. *Proceedings of the Second Workshop on Principles and Practice of Constraints Programming (PPCP'94)* Rosairo, Orcas Island, Washington, *Lecture Notes in Computer Science* **874** (Springer, Berlin, 1994) 326–335.
2. A. Aiken, D. Kozen, M. Vardi and E. Wimmers. The complexity of set constraints. *Proceedings of the 1993 Conference on Computer Science Logic (CSL'93)* Swansea, UK, *Lecture Notes in Computer Science* **832** (Springer, Berlin, 1993) 1–17. 1993.
3. F. Baader. Unification in commutative theories, Hilbert's basis theorem, and Gröbner bases. *Journal of the ACM* 40 (3) (1993) 477–503.
4. F. Baader and K.U. Schultz. Unification in the union of disjoint equational theories: combining decision procedures. *Proceedings of the Eleventh Conference on Automated Deduction (CADE-11),* Saratoga Springs, New York, *Lecture Notes in Artificial Intelligence* **607** (Springer, Berlin, 1992) 50–65.
5. F. Baader and J.S. Siekmann. Unification Theory. In: *Handbook of Logic in Artificial Intelligence and Logic Programming* (D.M. Gabbay, C.J. Hogger, J.A. Robinson, eds.), (Oxford, 1994).
6. H. Comon, M. Haberstrau, and J.-P. Jouannaud. Decidable problems in shallow equational theories. *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science (LICS)*, Santa Cruz, California, (IEEE Computer Society, Washington, D.C., 1992) 255–265.
7. N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. *Handbook of Theoretical Computer Science*, (Elsevier, Amsterdam, 1990) 245–320.
8. N. Dershowitz, S. Mitra, and G. Sivakumar. Decidable matching for convergent systems. *Proceedings of the Eleventh Conference on Automated Deduction (CADE--11),* Saratoga Springs, New York, *Lecture Notes in Artificial Intelligence* **607**, (Springer, Berlin, 1992) 589–602.
9. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* (Freeman, San Francisco, 1979)
10. W.D. Goldfarb. The undecidability of the second-order unification problem, *Theoretical Computer Science* **13** (1981) 225–230.
11. N. Heintze. Set-based analysis of ML programs. *Proceedings of the 1994 ACM Conference on Lisp and Functional Programming,* June 1994.
12. J.-M. Hullot. Canonical forms and unification. In: *Proceedings of the Fifth Conference on Automated Deduction (CADE-5),* Les Arcs, France, *Lecture Notes in Computer Science* **87**, (Springer, Berlin, 1980) 318–334.
13. J.-P. Jouannaud and C. Kirchner. Solving Equations in Abstract Algebras: a Rule--Based Survey of Unification. In: *Computational Logic: Essays in Honor of Alan Robinson,* (J.-L. Lassez and G.D. Plotkin, Eds.), (MIT, Cambridge, Mass., 1991) 360–394.
14. E. Kaltofen, M.S. Krishnamoorthy and B.D. Saunders. Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM Journal of Algebraic and Discrete Methods* 8 (4) October 1987, 683–690.
15. D. Kapur and P. Narendran. Complexity of unification problems with associative--commutative operators. *Journal of Automated Reasoning* **9** (2) (1992) 261–288.
16. D. Kapur and P. Narendran. NP-completeness of the set unification and matching problems. *Proceedings of the Eighth Conference on Automated Deduction (CADE-8),* Oxford, *Lecture Notes in Computer Science* **230** (Springer, Berlin, 1986) 489–495.

17. C. Kirchner. Computing unification algorithms. *Proceedings of the First Annual IEEE Symposium on Logic in Computer Science (LICS),* (IEEE Computer Society, Washington, D.C., 1986), 206–216.
18. A. Martelli and U. Montanari. An efficient unification algorithm *ACM Transactions on Programming Languages and Systems* Vol. **4**, No. **2** (ACM. 1982), 258–282.
19. P. Narendran, F. Pfenning and R. Statman. On the unification problem for Cartesian Closed Categories. *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science (LICS),* Montreal, Canada, (IEEE Computer Society, Washington, D.C., 1986). (To appear in the *Journal of Symbolic Logic.*)
20. W. Nutt. Unification in monoidal theories. Proceedings of the *10th International Conference on Automated Deduction (CADE-10),* Kaiserslautern, West Germany, July 1990 (Springer LNCS 449).
21. G.D. Plotkin. Building-in Equational Theories, in: *Machine Intelligence* **7**, (B. Meltzer and D. Michie, Eds.), (Edinburgh, 1972), 73–90.
22. T.J. Schaefer. The complexity of satisfiability problems. *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC),* (ACM, New York, 1978), 216–226.
23. M. Schmidt-Schauss. An algorithm for distributive unification. Interner Bericht 13/94, Fachbereich Informatik, Universität Frankfurt, Frankfurt, Germany, 1995.
24. D.A. Wolfram. *The Clausal Theory of Types.* Volume 21 of Cambridge Tracts in Theoretical Computer Science, (Cambridge, 1993).

A Appendix

Theorem 14. *Let $S = \{ (s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m) \}$ be a set of pairs of elementary terms where for every $(s_i, t_i) \in S$ both s_i and t_i are non-ground terms. The set S is ACN-unifiable iff S is $ACUN$-unifiable.*

Proof. One way is trivial; if S is ACN-unifiable, then it is clearly $ACUN$--unifiable. We now prove the other way, i.e. if S is $ACUN$-unifiable then S is ACN-unifiable.

Suppose θ is an $ACUN$-unifier for S. We define another substitution $\theta'(x_i) = \theta(x_i) + 0$ for every variable x_i in S.

For every $(s_i, t_i) \in S$, either $\theta s_i =_{ACN} s'_i$ or $\theta s_i =_{ACN} s'_i + 0$ where s'_i is a ground term without 0 and it is an ACN normal form. We note that nilpotence can be considered as a rule $N : x + x \rightarrow 0$. The rewriting system N is convergent modulo AC. The ACN normal form is the normal form with respect to $\rightarrow_{N/AC}$.

Similarly, either $\theta t_i =_{ACN} t'_i$ or $\theta t_i =_{ACN} t'_i + 0$ where t'_i is a ground term without 0 and it is an ACN normal form. Furthermore, $\theta s_i =_{ACUN} \theta t_i$ implies $s'_i =_{ACN} t'_i$.

Because s_i and t_i are non-ground terms, $\theta' s_i =_{ACN} s'_i + 0$ and $\theta' t_i =_{ACN} t'_i + 0$. Thus $\theta' s_i =_{ACN} \theta' t_i$. □