ACUNh: Unification and Disunification Using Automata Theory*

Pascal Lafourcade[1,2], Denis Lugiez[2] and Ralf Treinen[1]

[1] LSV, ENS de Cachan, CNRS UMR 8643 & INRIA Futurs
[2] LIF, Université-Marseille 1 & CNRS UMR 6166

Abstract. We show several results about unification problems in the equational theory ACUNh consisting of the theory of *exclusive or* with one homomorphism. These results are shown using only techniques of automata and combinations of unification problems.
We show how to construct a most-general unifier for ACUNh-unification problems with constants using automata. We also prove that the first-order theory of ground terms modulo ACUNh is decidable if the signature does not contain free non-constant function symbols, and that the existential fragment is decidable in the general case. Furthermore, we show a technical result about the set of most-general unifiers obtained for general unification problems.

1 Introduction

In this paper we are interested in unification, disunification, and more generally in deciding the first-order theory of terms modulo the equational theory ACUNh. This theory consists of the following equational axioms:

$$\begin{array}{ll}
\text{(A)} & x \oplus (y \oplus z) = (x \oplus y) \oplus z \\
\text{(C)} & x \oplus y = y \oplus x \\
\text{(U)} & x \oplus 0 = x \\
\text{(N)} & x \oplus x = 0 \\
\text{(h)} & h(x \oplus y) = h(x) \oplus h(y)
\end{array}$$

Our interest in these problems was raised by our recent work on the symbolic verification of cryptographic protocols modulo the equational theory ACUNh [DLLT06]. The result of that paper is a complete constraint solving algorithm for the particular kind of symbolic constraints that correspond to the existence of an attack against the security of a cryptographic protocol, taking into account some properties of the cryptographic primitives described by the equational theory ACUNh. The constraint solving algorithm proceeds by several successive simplification steps. The completeness of these steps relies on the notion of a *non-collapsing solution*: A solution σ to a constraint system $\mathcal{C}$ is *non-collapsing*

* This work has been partly supported by the RNTL project PROUVÉ (n° 03 V 360) and the ACI-SI Rossignol.

if $s\sigma \neq t\sigma$ for different terms s, t taken from some finite set derived from the constraint system $\mathcal{C}$.

In order to use completeness assertions for non-collapsing solutions to show overall completeness of our algorithm we had at one step to guess the equations between terms (how this is done is described in [DLLT06]), and for each guess of equations compute a finite and complete set of unifiers (this is subject of the present paper).

Furthermore, in the case of unification with free function symbols (general unification) we needed a technical lemma assuring that the most general unifiers obtained do not introduce "new structural elements" not already present in the constraint system. This was necessary since one of the early steps of our algorithm consisted in guessing a particular specialization of our constraint system by guessing from the structural elements present in the constraint system. The technical lemma assures us that the guessing of equalities and the subsequent application of the resulting unifiers does not invalidate our earlier choice of "structural elements". The definition of "structural elements" and the statement of the technical lemma will be made precise later in the paper.

Finally, our constraint solving algorithm for cryptographic protocols relies on a notion of *well-definedness*, which is in particular satisfied for all *deterministic* protocols. Determinism of a protocol can be expressed as a *dis*-unification problem in the equational theory ACUNh.

The equational theory ACUNh is one example of a *monoidial*, or more generally a *commutative* equational theory. There is a wealth of results on this class of equational theories and on particular theories from this class. Before recalling the existing results relevant in our context let us recall the classical syntactic hierarchy of E-unification problems [BS01]:

- *elementary* E-unification problems are systems of equations between terms built with functions symbols in E and variables;
- E-unification problems *with constants* are systems of equations where the terms are built with functions symbols from E, free constants, and variables;
- *general* E-unification problems are systems of equations of terms built from function symbols from E, free function symbols, and variables.

ACUNh-unifiability with constants has been shown in [GNW00] to be decidable in polynomial time (this problem has been called *elementary* unification there, in deviance from the now established terminology). Furthermore, that paper states NP-completeness of the general ACUNh-unification problem, referring to the Baader-Schulz combination technique [BS96] for the existence of an NP-algorithm, and for NP-hardness to the proof of NP-hardness of the similar theory ACUN in [GNW00].

Baader gives an algorithm for the unification of several equational theories that involve homomorphism, for instance Abelian groups [Baa93]. On the one hand the results obtained in this and subsequent papers are general in that they apply to a whole class of commutative theories. Their drawback, on the other hand, is that they rely on the machinery of Gröbner bases for solving equations over the semi-ring associated to an equational theory.

Some results obtained by general methods for unification problems in commutative theories (see [BS01]) useful in our context are:

- ACUNh is unitary for elementary unification. This has been shown in [Baa93] for the similar theory AGh (Abelian groups with a homomorphism). This proof should transfer immediately to our setting ACUNh. An independent proof is given in this paper.
- As a consequence, and due to the fact that the corresponding semi-ring $\mathbb{Z}/2\mathbb{Z}[h]$ is a ring, ACUNh is unitary for unification with constants [BS01].
- Again as a consequence, ACUNh is finitary for general unification [BS01].

In this paper we use an alternative proof technique based on automata theory and prove that even the complete first-order theory of terms modulo ACUNh with free constants (but without free non-constant functions symbols) is decidable. We obtain from our automata construction an alternative algorithm for computing a finite complete most general set of unifiers for unification with constants. Finally, we use combination techniques [BS96] to obtain algorithms for computing finite complete sets of most general unifiers for general ACUNh-unification, and to decide general ACUNh-disunification problems. The above mentioned result of general ACUNh-unification not introducing "new structural elements" is based on an analysis of the combination algorithm applied to our setting.

Decidability of the first-order theory of terms modulo ACUNh in presence of free function symbols remains open.

2 Preliminaries: Automatic Structures

The exposition of the general method of *automatic structures* follows [BG00] which is the first systematic investigation of this concept.

The basic idea is to first provide an encoding of elements of the structure by words, and then to construct for any formula an automaton that accepts exactly those words that encode a solution of the formula. For technical reasons, this construction is restricted two purely *relational* signatures, that is signatures which do not contain constant symbols of function sysmbols. Note that it is always possible to transform a structure into a structure over a relational signature: we just have to replace the constants by unary predicates, and replace all n-ary functions by $n+1$-ary relations.

Another technical problem consists in the fact that the set of solutions of a formula is not a set of values, but a set of n-tuples of values where n is the number of free variables of the formula. That is, we have to provide a way to extend our encoding of elements of the structure to encodings of tuples of elements.

Definition 1. *Let Σ be a finite alphabet and $\square \notin \Sigma$. The* convolution *of words $x_1, \ldots, x_n \in \Sigma^*$ is defined as*

$$x_1 \otimes \ldots \otimes x_n := \begin{bmatrix} x_1^1 \\ \vdots \\ x_n^1 \end{bmatrix} \ldots \begin{bmatrix} x_1^l \\ \vdots \\ x_n^l \end{bmatrix}$$

where $l = max\{|x_i| \mid i \leq i \leq n\}$ is the length of the longest word among the x_i, and

$$x_i^j = \begin{cases} \textit{the } j\textit{-th symbol of } x_i \textit{ if } j \leq |x_i| \\ \square \textit{ otherwise} \end{cases}$$

Definition 2. *Let $\mathcal{A}$ be a structure over a relational signature with relation symbols $R_1, \ldots, R_n$. An* automatic representation *of $\mathcal{A}$ is given by*

1. *a finite alphabet Σ.*
2. *a regular language $L_\delta \subseteq \Sigma^*$*
3. *a surjective function $\nu \colon L_\delta \to \mathcal{A}$*
4. *a regular language $L_R \subseteq (\Sigma \cup \{\square\})^*$ for every relation symbol R of the signature of $\mathcal{A}$, such that for all $x_1, \ldots, x_n \in L_\delta$:*

$$x_1 \otimes \ldots \otimes x_n \in L_R \textit{ iff } (\nu(x_1), \ldots, \nu(x_n)) \in R^{\mathcal{A}}$$

A structure having an automatic representation is called automatic*.*

Note that there may be several possible automata for a given atomic formula since the behaviour of the automaton is not specified when the input word is not in $L_\delta \otimes \cdots \otimes L_\delta$.

Theorem 1 ([BG00]). *Let $\mathcal{A}$ be a relational structure with an automatic representation. Then the theory of $\mathcal{A}$ is decidable.*

The probably best-known example of an automatic structure is Presburger arithmetic (see, for instance, [CDG$^+$97]).

3 Unification and Diophantine Equations

In this section we recall the relation between unification with constants in ACUNh and linear equation solving in the ring $\mathbb{Z}/2\mathbb{Z}[h]$. This is in fact an instance of the by now classical connection between E-unification with constants for monoidial equational theories and linear equation solving over the associated semi-ring (see, for instance, [Nut90]). This section just serves as a reminder of some basic and well-known results used in the following sections.

In the rest of the paper, we use some notations that are useful to deal with terms and polynomials of $\mathbb{Z}/2\mathbb{Z}[h]$. The multiplication between polynomials p and q is denoted by $p \cdot q$. A polynomial $p(h) \in \mathbb{Z}/2\mathbb{Z}[h]$ can be written $\sum_{i=0}^n b_i h^i$ where $b_i \in \mathbb{Z}/2\mathbb{Z}$. The product $\odot$ of a polynomial by a term is a term defined as follows:

$$(\sum_{i=0}^n b_i h^i) \odot t = \sum_{i=0 \mid b_i \neq 0}^{n} h^i(t)$$

For instance $(h^2 + 1) \odot (X \oplus h(a)) = h^2(X) \oplus X \oplus h^3(a) \oplus h(a)$. Conversely, a term t such that $\mathcal{V}(t) = \{X_1, \ldots, X_p\}$ can be written $t^{X_1} \odot X_1 \oplus \ldots \oplus t^{X_p} \odot X_p \oplus t_0$ for some $t^{X_1}, \ldots, t^{X_p} \in \mathbb{Z}/2\mathbb{Z}[h]$, and t_0 a ground term.

Note that we use the symbol $+$ for the addition operation in $\mathbb{Z}/2\mathbb{Z}[h]$, while $\oplus$ is the binary operator of the term algebra.

We denote by $deg(p)$ the degree of a polynomial, that is $deg(\sum_{i=0}^n b_i h^i) = n$ in case $b_i \neq 0$. By extension, $deg(p_1, \ldots, p_n) = (deg(p_1), \ldots, deg(p_n))$.

3.1 Linear Diophantine Equations in $\mathbb{Z}/2\mathbb{Z}[h]$

Let (HE) be a homogeneous system of equations of the following form:

$$\begin{cases} A_{1,1} \cdot X_1 + \ldots + A_{1,n} \cdot X_n = 0 \\ \ldots \\ A_{m,1} \cdot X_1 + \ldots + A_{m,n} \cdot X_n = 0 \end{cases} \tag{HE}$$

where the $A_{i,j}$'s are polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$, and the unknowns take values in $\mathbb{Z}/2\mathbb{Z}[h]$. We denote by $Sol(HE)$ the set of solutions to (HE).

Definition 3. *We define a quasi-order on* $(\mathbb{Z}/2\mathbb{Z}[h])^m$ *by*

$$(p_1, \ldots, p_m) \lesssim (q_1, \ldots, q_m) \Leftrightarrow \forall 1 \leq i \leq m : \quad deg(p_i) \leq deg(q_i)$$

The pertaining strict order is derived from this as usual by

$$\begin{aligned}(p_1, \ldots, p_m) < (q_1, \ldots, q_m) \Leftrightarrow {} & (p_1, \ldots, p_m) \lesssim (q_1, \ldots, q_m) \\ & \textit{and not } (q_1, \ldots, q_m) \lesssim (p_1, \ldots, p_m)\end{aligned}$$

For instance, $(h^2, 1) < (h^2, h^3)$ and $(h^3 + h, h^2) \lesssim (h^3, h^2 + 1)$, while (h^2, h^3) and (h, h^4) are not comparable by the quasi order $\lesssim$. Note that the strict order $<$ is well-founded.

From this quasi-order we derive an equivalence relation as usual:

Definition 4. *We define an equivalence relation on* $(\mathbb{Z}/2\mathbb{Z}[h])^m$ *by*

$$\begin{aligned}(p_1, \ldots, p_m) \sim (q_1, \ldots, q_m) \Leftrightarrow {} & (p_1, \ldots, p_m) \lesssim (q_1, \ldots, q_m) \\ & \textit{and } (q_1, \ldots, q_m) \lesssim (p_1, \ldots, p_m)\end{aligned}$$

In other words, $(p_1, \ldots, p_m) \sim (q_1, \ldots, q_m)$ iff $deg(p_i) = deg(q_i)$ for each i. Note that the equivalence classes of $\sim$ are uniquely identified by m-tuples of integers (the vector of degrees of the polynomials), and that every equivalence class is finite (since the coefficients are in $\{0, 1\}$).

Fact 1 *The number of minimal solutions to a system of equations (HE) is finite.*

Proof. We recall Dickson's classical lemma [Dic13]: every infinite sequence of distinct tuples of natural numbers contains at least two (actually infinitely many) comparable tuples.

Let us assume that the number of minimal solutions to (HE) is infinite. This yields an infinite sequence $\boldsymbol{T_1}, \boldsymbol{T_2}, \ldots$ of distinct incomparable tuples of polynomials. Therefore we would have an infinite sequence of incomparable tuples of natural numbers $deg(\boldsymbol{T_1}), deg(\boldsymbol{T_2}), \ldots$, in contradiction to Dickson's lemma. □

Fact 2 *Every solution to (HE) is a linear combination (with coefficients in* $\mathbb{Z}/2\mathbb{Z}[h]$*) of minimal solutions to (HE).*

Proof. Let σ be a non-minimal solution to (HE). Since $<$ is well-founded there exists a minimal solution τ to (HE) with $\tau < \sigma$. Let

$$d = min\{deg(\sigma_i) - deg(\tau_i) | 1 \leq i \leq m\}$$

We define σ' by

$$\sigma'(X_i) = \sigma(X_i) - h^d \cdot \tau(X_i) \quad \text{for all } i,\ 1 \leq i \leq m$$

which obviously is again a solution to (HE) since the set of solutions is closed under multiplication with scalars and under sums. Furthermore, by the choice of d, $\sigma' < \sigma$. The claim follows by induction. □

3.2 From Linear Equations over $\mathbb{Z}/2\mathbb{Z}[h]$ to ACUNh-Unification with Constants

The rest of this section is devoted to the construction of a most general unifier for a given ACUNh-unification problem with constants. As a consequence of the construction, ACUNh is unitary for unification with constants.

Let $\Sigma_C = \{c_1, \ldots, c_k\}$ be a given finite set of free constant symbols. We consider a unification problem, i.e. a conjunction of equations $s_j = t_j$ for $j = 1, \ldots, m$ where s_j, t_j are terms containing free constants from Σ_C, the homomorphism symbol h, the binary operator $\oplus$, and the constant 0. Let $x_1, \ldots, x_n$ be the variables occurring in the unification problem. Using the notation introduced in the previous section and the algebraic properties of $\oplus$, we get that the unification problem is equivalent to a system of equations (U):

$$\sum_{i=1}^{i=n} A_{i,j} \odot x_i = b_j \quad \text{for} \quad j = 1, \ldots, m \tag{U}$$

where $A_{i,j}$ are polynomials of $\mathbb{Z}/2\mathbb{Z}[h]$, the b_j are ground terms, and where the variables x_i range over terms. Let (HU) be the equation system obtained from (U) by replacing all the right hand sides by the term 0:

$$\sum_{i=1}^{i=n} A_{i,j} \odot x_i = 0 \quad \text{for} \quad j = 1, \ldots, m \tag{HU}$$

We denote by $Sol(U)$ (resp. $Sol(HU)$) the set of ground substitutions that are solutions to (U) (resp. (HU)).

Fact 3 *For any $\sigma \in Sol(U)$ we have $Sol(U) = \sigma \oplus Sol(HU)$*

Proof. This follows immediately from the properties ACUNh. □

An arbitrary ground solution of (U) can be obtained as follows: Each of the terms occurring on the right-hand side of (U) can be decomposed as

$$b_j = \sum_{i=1}^{i=k} B_j^i \odot c_i$$

For any $i = 1, \ldots, k$, let (E_i) be the equation system

$$\sum_{i=1}^{i=n} A_{i,j} \cdot X_i = B_j^i \quad \text{for} \quad j = 1, \ldots, m \tag{E_i}$$

where the variables X_i range over polynomials from $\mathbb{Z}/2\mathbb{Z}[h]$. If σ_i is a solution to (E_i) for each $i = 1, \ldots, k$, then we obtain a solution σ to (U) by

$$\sigma(x_j) = \sum_{i=1}^{i=k} \sigma_i(X_j) \odot c_i$$

A most-general unifier of (HU) is obtained as follows: Let (HE) be the equation system

$$\sum_{i=1}^{i=n} A_{i,j} \cdot X_i = 0 \quad \text{for} \quad j = 1, \ldots, m \tag{HE}$$

where variables X_i range over polynomials from $\mathbb{Z}/2\mathbb{Z}[h]$. By Fact 1, the system (HE) has a finite set of minimal solutions $\{\sigma_1, \ldots, \sigma_\mu\}$. In the follwoing we denote $I_\mu = \{1, \ldots, \mu\}$.

Fact 4 *The homogeneous unification problem (HU) has a most general unifier σ_H defined by $x_i\sigma_H = \Sigma_{k \in I_\mu} P_{i,k} \odot y_k$ where $\sigma_k = \{X_1 \leftarrow P_{1,k}, \ldots, X_n \leftarrow P_{n,k}\}$ with $P_{i,k} \in \mathbb{Z}/2\mathbb{Z}[h]$, and where the y_k are fresh variables.*

Proof.

- Firstly we prove that σ_H is a solution of (HU). For $j = 1, \ldots, m$, we have:

$$\begin{aligned}(\Sigma_{i=1}^{i=n} A_{i,j} \odot x_i)\sigma_H &= \Sigma_{i=1}^{i=n} A_{i,j} \odot (x_i\sigma_H) \\ &= \Sigma_{i=1}^{i=n} A_{i,j} \odot (\Sigma_{k \in I_\mu}(P_{i,k} \odot y_k)) \\ &= \Sigma_{i=1}^{i=n} \Sigma_{k \in I_\mu}(A_{i,j} \odot (P_{i,k} \odot y_k)) \\ &= \Sigma_{i=1}^{i=n} \Sigma_{k \in I_\mu}(A_{i,j} \cdot P_{i,k}) \odot y_k) \\ &= \Sigma_{k \in I_\mu}((\Sigma_{i=1}^{i=n} A_{i,j} \cdot P_{i,k}) \odot y_k) \\ &= 0\end{aligned}$$

- Then we prove that any solution σ of (HU) is an instance of σ_H.
  Let $\mathcal{Z}$ be the set of variables occuring in $x_i\sigma$. Since these variables are no longer instantiated, they are treated as constants in the following. For $i = 1, \ldots, n$ we have $x_i\sigma = (\Sigma_{c \in \Sigma_C} X_i^c \odot c) \oplus (\Sigma_{z \in \mathcal{Z}} Z_i^z \odot z)$ and $(x_1\sigma, \ldots, x_n\sigma)$ is a solution of (HU) iff for each $c \in \Sigma_C$, for each $z \in \mathcal{Z}$ we have that $(X_1^c, \ldots, X_n^c)$ and $(Z_1^z, \ldots, Z_n^z)$ are solutions of (HE).
  Therefore for each $c \in \Sigma_C$, $(X_1^c, \ldots, X_n^c)$ is a linear combination of the minimal solution of (HE), i.e. $X_i^c = \Sigma_{k \in I_\mu} Q_k^c \cdot P_{i,k}$ for $i = 1, \ldots, n$ where the Q_k^c's are the coefficients of the linear combination. For each $z \in \mathcal{Z}$, the same holds for $(Z_1^z, \ldots, Z_n^z)$'s, yielding $Z_i^z = \Sigma_{k \in I_\mu} R_k^z \cdot P_{i,k}$.
  Therefore for $i = 1, \ldots, n$,

$$\begin{aligned}x_i\sigma &= (\Sigma_{c \in \Sigma_C}(\Sigma_{k \in I_\mu} Q_k^c \cdot P_{i,k}) \odot c) \oplus (\Sigma_{z \in \mathcal{Z}}(\Sigma_{k \in I_\mu} R_k^z \cdot P_{i,k}) \odot z) \\ &= \Sigma_{k \in I_\mu}((\Sigma_{c \in \Sigma_C} P_{i,k} \odot (Q_k^c \odot c)) \oplus (\Sigma_{z \in \mathcal{Z}} P_{i,k} \odot (R_k^z \odot z))) \\ &= \Sigma_{k \in I_\mu}(P_{i,k} \odot (\Sigma_{c \in \Sigma_C}(Q_k^c \odot c)) \oplus (P_{i,k} \odot \Sigma_{z \in \mathcal{Z}}(R_k^z \odot z))) \\ &= \Sigma_{k \in I_\mu} P_{i,k} \odot (\Sigma_{c \in \Sigma_C} Q_k^c \odot c \oplus \Sigma_{z \in \mathcal{Z}} R_k^z \odot z)\end{aligned}$$

which terminates the proof (choose $y_k = (\Sigma_{c\in\Sigma_C} Q_k^c \odot c) \oplus (\Sigma_{z\in\mathcal{Z}} R_k^z \odot z)$). □

Fact 5 *Let σ be a ground solution to (U) and σ_H a most-general unifier of (HU). The substitution $\sigma \oplus \sigma_H$ is a most-general unifier of (U).*

Proof. This follows from Facts 3 and 4 since σ is ground. □

Fact 6 *The theory ACUNh is unitary for unification with constants.*

Proof. This follows from Fact 5. □

4 Finding Minimal Solutions of Systems of Equations Using Automata

What we need is a way to compute the minimal solutions to a homogeneous system of Diophantine equations (HE), and to compute some (small) solution to an inhomogeneous system (E). One possible approach is to perform algebraic computations similar to what is done by AC-unification algorithms. Instead, we shall use an automata-theoretic approach that yields a more general result:

Let $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, +, 0, h\rangle$ denote the structure consisting of the universe $\mathbb{Z}/2\mathbb{Z}[h]$ with the relation $\lesssim$ and the operations $+$, 0, and h. We show that the first-order theory of this structure is decidable since it is an *automatic structure* [BG00].

Lemma 1. *The first-order theory of $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, +, 0, h\rangle$ is decidable.*

Proof. We show that the structure is automatic [BG00].

A polynomial $p(h) = \sum_{i=1}^{i=n} b_i h \in \mathbb{Z}/2\mathbb{Z}[h]$, where $b_i = 1$, is represented by the word $\nu(p) = b_0 \cdots b_n$ (that is, the least significant bit first). The polynomial 0 is represented by the the word 0. The image of ν is described by the regular expression $0 \cup 0^*1$ and hence recognizable.

We now give automata accepting the representations of tuples of polynomials that are in the three relations of the structure $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, +, 0\rangle$ (We have to replace the constant 0 by a unary relation $X_1 = 0$, and the function h by a relation $X_1 = h(X_2)$). The general construction as explained in Section 2 requires usage of a padding symbol □ when representing tuples of values. In the case of arithmetic, we can for the sake of simplicity just replace the symbol □ by the symbol 0.

The automaton for $X_1 = 0$ is trivial and omitted. The automaton for $X_1 \lesssim X_2$ is given in Figure 1.

The automaton for $X_1 = X_2 + X_3$ is given in Figure 2. Note that the automaton is simpler than the automaton for the addition of Presburger arithmetic since there is no carry-over to deal with.

The automaton accepting the pairs (X_1, X_2) such that $X_1 = h(X_2)$ is more complex: it contains two states which remember the previous values of X_2. It is described in Figure 3. □

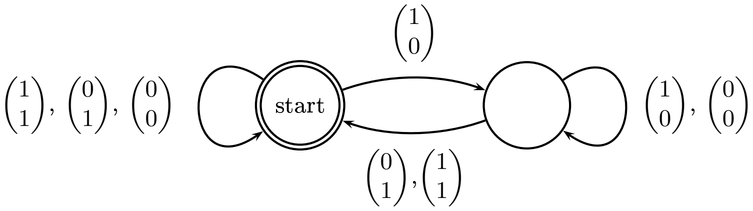
start
$\binom{1}{0}$
$\binom{1}{1}, \binom{0}{1}, \binom{0}{0}$
$\binom{1}{0}, \binom{0}{0}$
$\binom{0}{1}, \binom{1}{1}$

Fig. 1. Automata for $X_1 \lesssim X_2$.

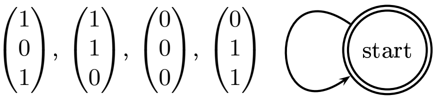
start
$\begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}$

Fig. 2. Automata for $X_1 = X_2 + X_3$.

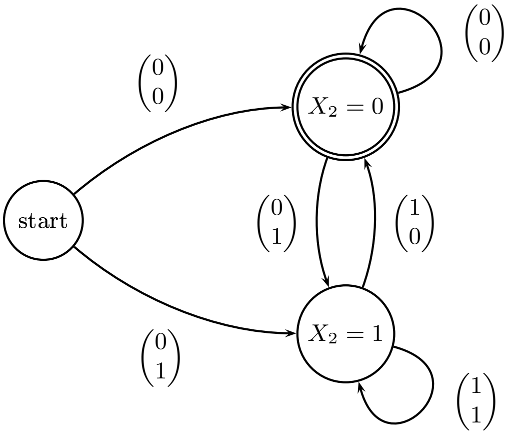

Fig. 3. Automaton for $X_1 = h(X_2)$

We can now easily obtain a small ground solution for an inhomogeneous system of linear equations by constructing the automaton for the equation system, and reading off a short solution (for instance one which does not pass twice by the same state). We also obtain the minimal solutions to a homogeneous system of Diophantine equations (HE).

Fact 7 *The set of minimal solutions to a homogeneous system of linear Diophantine equations is computable.*

Proof. A vector $\boldsymbol{X}$ is a minimal solution to a system of Diophantine equations $\phi(\boldsymbol{X})$ iff it is a solution of the following formula:

$$\phi(\boldsymbol{X}) \wedge \forall \boldsymbol{Y} \left(\boldsymbol{Y} \lesssim \boldsymbol{X} \wedge \phi(\boldsymbol{Y}) \rightarrow \boldsymbol{X} \lesssim \boldsymbol{Y} \right)$$

The set of elements $\boldsymbol{X}$ which satisfy this formula is accepted by an automaton which is effectively computable. The language of this automaton is finite since there is only a finite number of minimal solutions. To obtain the set of minimal solutions, we simply use the automaton to generate all the terms of its language. □

5 General ACUNh-Unification

5.1 A Unification Algorithm

To apply the combination algorithm of [BS96], we must prove that unification with linear constant restriction is finitary. Given a unification problem (*i.e.* a finite set of equations $s_i = t_i$), we associate to each constant c appearing in the problem a set V_c of variables that are the variables in which c must not occur.

Assume that we have a linear ordering $<$ on the set of constants Σ_C and variables $\mathcal{X}$, then we define $V_c = \{x \in \mathcal{X} \mid x < c\}$. A unification problem with linear constant restriction is a unification problem with the additional constraint restriction corresponding to the given ordering $<$. This amounts to stating that each variable x of the problem can be instantiated only by terms containing constants c such that $x \notin V_c$. This set is computable and finite and we can write $x = \Sigma_{\{x \notin V_c\}} X_{i,c} \odot c$ for $X_{i,c}$ a polynomial of $\mathbb{Z}/2\mathbb{Z}[h]$. Therefore unification problems with linear constant restriction are solved in the same way as unification problems are.

As a result, we get a unification algorithm for the theory ACUNh in Σ extended with free symbols as a simple application of the combination algorithm (actually we can even choose the simpler version designed for the combination with the empty theory, see [BS96]).

5.2 A Technical Result about Unification

To prove the next result (Lemma 2), we shall rely on notations and algorithms introduced in the study of combination algorithms, see [BS96] for more details.

From now on, we assume that $\mathcal{F} = \Sigma \uplus \Sigma'$ where Σ' is a set of free symbols which contains at least one symbol of arity greater than or equal to 2. The context notation is extended as follows: $t = C[t_1, \ldots, t_n]$ if C is a context made of symbols of Σ only and the t_i's do not have a symbol from Σ at their root, or if C is a context made of symbols of Σ' and the t_i's do not have a symbol from Σ' at their root.

If a term t contains only symbols of Σ and variables, or only symbols of Σ' and variables we say that it is *pure*.

The number of theory alternation in a term is defined by $\#(t) = 0$ if t is pure, otherwise $\#(C[t_1, \ldots, t_n]) = 1 + max\{\#(t_i) \mid i = 1, \ldots, n\}$.

Definition 5. *The set $AF(t)$ of* alien factors *of t is defined by:*

- *$AF(t) = \{t\}$ if t is pure,*
- *$AF(t = C[t_1, \ldots, t_n]) = \{t\} \cup AF(t_1) \cup \ldots \cup AF(t_n)$*

Definition 6. *The set $St_{\mathsf{E}}(t)$ of* subterms *of t is the smallest set such that:*

- *$0, t \in St_{\mathsf{E}}(t)$,*
- *if $f(t_1, \ldots, t_n) \in St_{\mathsf{E}}(t)$ with $f \in \Sigma'$ then $t_1, \ldots t_n \in St_{\mathsf{E}}(t)$,*
- *if $s = f(t_1, \ldots, t_n) \in St_{\mathsf{E}}(t)$ with $f \in \Sigma$ then $AF(s) \subseteq St_{\mathsf{E}}(t)$.*

Example 1. Let $t_1 = h^2(a) \oplus b \oplus x$ and $t_2 = h(\langle a, b\rangle) \oplus x$, we get $St_{\mathsf{E}}(t_1) = \{t_1, a, b, x\}$ and $St_{\mathsf{E}}(t_2) = \{t_2, \langle a, b\rangle, a, b, x\}$.

Lemma 2. *Let P be a general (that is, including free function symbols) unification problem in the theory $\mathsf{E} = \mathsf{ACUNh}$ and θ be an mgu_{E} of P. Then for all $x \in dom(\theta)$ and $v \in St_{\mathsf{E}}(x\theta) \setminus \mathcal{V}(x\theta)$ there exists $t \in St_{\mathsf{E}}(P)$ such that $v =_{\mathsf{E}} t\theta$.*

Actually, we prove the result for the complete set of unifiers computed by the combination algorithm described by Baader and Schulz in [BS96].

Proof. Firstly, we remark that the lemma is true for a pure unification with linear constant restriction. This is obvious for the empty theory, and for ACUNh it is a consequence of our results on unification: a solution of a system of equations $\bigoplus_{i\in I} P_i(h) \odot x_i \oplus \bigoplus_{j\in J} Q_j(h) \odot c_j = 0$ with linear constant restriction is a linear combination of fresh variables and c_i's.

To generalize to the union of the theories, we analyze the combination algorithm. We recall this (non-deterministic) algorithm.

(1) Replace each non pure term $t = C[t_1, \ldots, t_n]$ by $C[x_{t_1}, \ldots, x_{t_1}]$ and add the equations $x_{t_i} = t_i$ where the x_{t_i}'s are fresh variables.
(2) Replace each equation $s = t$ such that s, t are pure but not in the same theory by $x_{s,t} = t \wedge x_{s,t} = s$ where $x_{s,t}$ is a new variable.
(3) Choose a partition of the set of variable $\mathcal{X}_1, \ldots, \mathcal{X}_p$, for each $\mathcal{X}_i$ choose a representative x_i and replace all variables $x \in \mathcal{X}_i$ by x_i (this amounts to adding equations $x_i = x$ for all $x \in \mathcal{X}_i$).
(4) Label each variable by Σ or Σ' non-deterministically, and choose a linear ordering $x_1 < \ldots < x_n$.

(5) The problem is decomposed into two pure unification problems with linear constant restrictions (otherwise return fail). Each problem is solved by taking the variable of the other theories as constant and the variables of the theory as variables. The unifier is given by the combination of the solutions of both unification problems (some replacement can be done to get the actual substitution).

We use the following properties of the algorithm. Assume that the algorithm returns the substitution θ.

- For each pair of variables x, y in the same equivalence class $x\theta = y\theta$.
- For each alien factor $t = C[t_1, \dots, t_n]$ of P, there exist variables $x_t, x_{t_1}, \dots, x_{t_n}$ such that $x_t\theta = t\theta = C[x_{t_1}\theta, \dots x_{t_n}\theta]$.
- For variable $x_{s,t}$, we have $x_{s,t}\theta = s\theta = t\theta$.
- For each term $C[x_1, \dots, x_l]$ occurring in the final pure unification problems, there exist y_{t_1} in the same equivalence class as x_1,..., y_{t_l} in the same equivalence class as x_l such that $C[t_1, \dots, t_l]$ is an alien factor of P.

The solution of the pure unification problems has the form: $x = C'[x_1, \dots, x_n]$ or x is a linear combination of fresh variables and variables x_i's and constants of Σ. In any case the factors of $x\theta$ for a variable x of the initial problems are either $X\theta$, or are some $x_t\theta$ for a variable x_t hence there are some $t\theta$ for a factor t of P or fresh variables. □

Actually, the combination algorithm computes a complete finite set of unifiers. To find the actual most general and minimal set of unifiers, one must add a last step which detects the unifiers that are subsumed by other unifiers. This step does not change the result and it is irrelevant for our purpose, since what is required in our result is that all possible ground substitutions covered by the set of unifiers that we consider.

6 Disunification and Beyond

6.1 General Disunification

We now turn to disunification problems, that is the problem of deciding the existential fragment of the first-order theory. In the following lemma we denote by $=_{AC}$ the equality relation modulo the axioms of associativity and commutativity. Let us recall that $=$ denotes equality modulo the axioms ACUNh.

Lemma 3. *Let $t_1, s_1, \dots, t_n, s_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ terms in normal form. Then the following two assertions are equivalent:*

1. *There exists no ground substitution σ with $t_1\sigma \neq s_1\sigma \wedge \dots \wedge t_n\sigma \neq s_n\sigma$*
2. *$t_i =_{AC} s_i$ for some $i \in \{1, \dots, n\}$*

Proof. **(2)⇒(1):** If $t_i =_{AC} s_i$ for some i then $t_i = s_i$ and $t_i\sigma = s_i\sigma$ for all substitutions. Hence, there exists no substitution σ with $t_1\sigma \neq s_1\sigma \wedge \ldots \wedge t_n\sigma \neq s_n\sigma$.

(1)⇒(2): Let us assume that (1) does not hold. Let $T = \{s_1, \ldots, s_n, t_1, \ldots, t_n\}$. If $t_i \neq_{AC} s_i$ for all $i, 1 \leq i \leq n$, then we proceed by induction on the cardinality n of the set $\mathcal{V}(T)$ of free variables of T.

- If $n = 0$ then the s_i and the t_i are ground. Hence, the empty substitution ϵ satisfies that $t_1\epsilon \neq s_1\epsilon \wedge \ldots \wedge t_n\epsilon \neq s_n\epsilon$
- If $n > 0$ then let $x \in \mathcal{V}(T)$, and let g be some ground term that
  1. is different from 0
  2. does not containing the symbol +
  3. is not a syntactic subterm of a term in T

  Let $t'_i = t_i[x \mapsto g]$ and $s'_i = s_i[x \mapsto g]$ for $1 \leq i \leq n$. We have that $t'_i \neq_{AC} s'_i$ for all i, and all the t'_i and s'_i are terms in normal form. Since $\{t'_1, \ldots, t'_n, s'_1, \ldots, s'_n\}$ contains $n-1$ variables there exists by induction hypothesis a substitution σ' such that

  $$t'_1\sigma' \neq s'_1\sigma' \wedge \ldots \wedge t'_n\sigma' \neq s'_n\sigma'$$

  Hence, setting $\sigma = \sigma' \circ [x \mapsto g]$ we obtain that

  $$t_1\sigma \neq s_1\sigma \wedge \ldots \wedge t_n\sigma \neq s_n\sigma$$ □

Theorem 2. *The existential fragment of the first-order theory of terms modulo the equational theory* ACUNh *is decidable.*

Proof. Given a closed existential formula $\phi = \exists\bar{x}\psi$, where ψ is a quantifier-free formula, let $c_1 \vee \ldots \vee c_n$ be a disjunctive normal form of ψ. Validity of ϕ is equivalent to validity of some $\exists\bar{x}c_i$.

Let

$$c = (r_1 = u_1 \wedge \ldots \wedge r_m = u_m \wedge s_1 \neq t_1 \wedge \ldots \wedge s_n \neq t_n)$$

This formula is satisfiable if there exists a most general unifier μ of

$$r_1 = u_1 \wedge \ldots \wedge r_m = u_m$$

such that the following formulas is satisfiable:

$$s_1\mu \neq t_1\mu \wedge \ldots \wedge s_n\mu \neq t_n\mu$$

There is a finite set of most general unifiers μ which can be computed, and satisfiability of the disequations is decidable due to Lemma 3. □

6.2 The First-Order Theory with Constants

Theorem 3. *The first order theory of terms over* ACUNh *with finitely many free constants is decidable.*

Proof. This follows immediately from Lemma 1 since the algebra of ground terms modulo the equational theory ACUNh with m free constants is isomorphic to the m-fold direct product of $\langle \mathbb{Z}/2\mathbb{Z}[h], +, h, 0\rangle$. □

7 Conclusions

We have shown that the first-order theory of ground terms modulo ACUNh is decidable if the signature contains only the symbols from ACUNh and free constant symbols, and that the existential fragment of this first-order theory is decidable for arbitrary signatures. The obvious question whether the complete first-order theory is decidable in the general case remains open.

As a consequence of the fact that the first-order theory of $\langle \mathbb{Z}/2\mathbb{Z}[h], +, h, 0 \rangle$ is automatic, and by the nature of the isomorphism between the m-fold product of this structure and the algebra of ground terms module ACUNh with m free constants, it follows that the latter structure is itself automatic. This result does not seem to extend to the general case: The natural extension to free function symbols would consist in using tree automata with component-wise equality tests. Unfortunately, this class of tree automata has an undecidable emptiness problem [SAN$^+$05], and is of no help in establishing decidability results.

References

[Baa93] Franz Baader. Unification in commutative theories, Hilbert's basis theorem and Gröbner bases. *Journal of the ACM*, 40(3):477–503, 1993.

[BG00] A. Blumensath and E. Grädel. Automatic structures. In *Proc. 15th IEEE Symposium on Logic in Computer Science (LICS'00)*, pages 51–62, Santa Barbara, California, USA, 2000. IEEE Comp. Soc. Press.

[BS96] Franz Baader and Klaus U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. Symbolic Computation*, 21:211–243, 1996.

[BS01] F. Baader and W. Snyder. Unification theory. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 8, pages 445–532. Elsevier Science, 2001.

[CDG$^+$97] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.

[Dic13] L. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n prime factors. *American Journal Mathematical Society*, 35:413–422, 1913.

[DLLT06] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06)*, Lecture Notes in Computer Science, Venice, Italy, July 2006. Springer. To appear.

[GNW00] Q. Guo, P. Narendran, and D. A. Wolfram. Complexity of nilpotent unification and matching problems. *Information and Computation*, 162(1-2):3–23, 2000.

[Nut90] Werner Nutt. Unification in monoidal theories. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 618–632, Kaiserslautern, Germany, July 1990. Springer-Verlag.

[SAN$^+$05] Zhendong Su, Alexander Aiken, Joachim Niehren, Tim Priesnitz, and Ralf Treinen. The first-order theory of subtyping constraints, 2005. Accepted for publication in ACM TOPLAS with minor revisions.