S. Delaune, P. Lafourcade,
D. Lugiez, R. Treinen

Symbolic Protocol Analysis in Presence of a Homomorphism Operator and Exclusive Or

Research Report LSV-05-20
November 2005

Laboratoire
Spécification
et
Vérification

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Symbolic Protocol Analysis in Presence of a Homomorphism Operator and *Exclusive Or* ·

Stéphanie Delaune[1,2], Pascal Lafourcade[2,3], Denis Lugiez[3] and Ralf Treinen[2]

[1] France Télécom, Division R&D
[2] LSV, CNRS UMR 8643, ENS de Cachan & INRIA Futurs
[3] LIF, Université Aix-Marseille1 & CNRS UMR 6166

Abstract. The symbolic verification of the security property of a cryptographic protocol for a bounded number of sessions is usually expressed as a symbolic trace reachability problem. Such a problem can be expressed as a constraint system for deducibility constraints for a certain inference system describing the possible actions of an attacker.
We show that symbolic trace reachability for well-defined protocols is decidable in presence of both the *exclusive or* operator and a homomorphism over this operator. The *exclusive or* operator is often used in security protocols as a symmetric encryption operation. The homomorphism may model a hash function, or may be used to model a special situation in asymmetric encryption where an intruder may encrypt a message but can never learn about the corresponding decryption key.
One main step of our proof consists in reducing the constraint system for deducibility into a constraint system for deducibility in one step and using one particular rule of the constraint system. This constraint system, in turn, can be expressed as a system of quadratic equations of a particular form over the ring of polynomials in one indeterminate over the finite field $\mathbb{Z}/2\mathbb{Z}[h]$. We show that satisfiability of these systems of equations is decidable.

1 Introduction

Cryptographic protocols are small programs designed to ensure secure communication via a channel that is controlled by an attacker. They involve a high level of concurrency and are difficult to analyze by hand. For instance, a flaw in the so-called Needham-Schroeder public key authentication protocol [NS78] was discovered by Lowe [Low96] using an automatic tool only 17 years after its publication. Therefore, the need of formal methods to achieve this analysis has been recognized and many approaches have been proposed.

A cryptographic protocol is defined by a set of programs (or *roles*) which may be executed by agents which are distributed over a network. In the simplest of cases these programs are linear sequences of *receive* and *send* instructions on a public communication channel. The attacker may modify the messages sent on

· This work has been partly supported by the RNTL project PROUVÉ 03V360 and the ACI-SI Rossignol.

the channel using a certain set of *intruder capabilities*. The fact that all messages may be modified by the attacker is often expressed by saying that *the attacker is the network*.

The most basic property of cryptographic protocols is the so-called *security* property, which states that for any number of agents executing the roles, for any possible interlacing of the program execution, and for any modifications of the messages by the attacker (according to his deduction capabilities) the intruder is not able to deduce a certain message which is supposed to remain secret.

Security of a protocol is undecidable if the number of role instances running in parallel is unbounded [DLMS99]. Rusinowitch and Turuani [RT03] have shown that security of a cryptographic protocol is decidable, and in fact NP-complete, when the number of parallel role instances is bounded. In this case there is only a bounded number of symbolic traces, each of which represents an interlacing of the execution of the parallel role instances. Every message received during the execution of a role is a message that can be deduced using the intruder deduction capabilities from the messages sent before on the communication channel. The idea of the algorithm is to guess a symbolic trace in which the messages are represented by terms containing variables. This symbolic trace corresponds to a concrete execution trace if the variables can be instantiated in such a way that at every moment a message received by an agent can in fact be deduced by the intruder from the messages seen before. Hence, verifying security of a protocol amounts to a non-deterministic guessing of the symbolic trace plus the resolution of a system of *deducibility constraints*. These constraints can be solved in NP in case of the intruder deduction capabilities described by the so-called *Dolev-Yao* model.

This result, as many others, relies on the so-called *perfect cryptography assumption* which states that the cryptographic primitives like encryption, hashing, etc are perfect and can be treated as black boxes. This assumption allows to represent messages built using cryptographic primitives in a free term algebra. On the one hand, even this quite strong assumption provides a useful abstraction of the problem since it allows one to detect many attacks on protocols, as for instance the attack on the Needham-Schroeder protocol mentioned before. On the other hand, many security flaws of protocols may go undetected when one abstracts away all properties of the cryptographic primitives. Indeed, some attacks exploit in a clever way the interaction between protocol rules and properties of cryptographic primitives.

Hence, a recent research direction in symbolic protocol analysis consists in relaxing the perfect cryptography assumption by taking into account some of the properties of the cryptographic primitives (see [CDL05] for a survey). The most important of these, the so-called *algebraic properties*, can be stated by a set of equational axioms. Most of these works extend the pioneering result by Rusinowitch and Turuani and take into account a particular (or a class of) equational theories [MS05,DJ04]. Recently, Chevalier and Rusinowitch [CR05] have developed a procedure to combine disjoint intruder theories, provided that satisfiability of intruder constraints can be decided in each theory.

In this paper, we prove the decidability of security protocols in *the active case* (the intruder may send messages that he has forged) for the equational theory ACUNh which is the combination of the homomorphism axiom $h(x+y) = h(x) + h(y)$ with the well-known *exclusive or* (ACUN) theory. Note that each of these two equational theories models basic properties of important cryptographic primitives. *Exclusive or* is a basic building block in many symmetric encryption methods like DES or AES. It is even used directly as an encryption method (Vernam encryption). Homomorphisms are ubiquitous in cryptography. Some protocols relying on these algebraic properties are described in [CDL05]. For instance, the ElGamal encryption method has this property. It also occurs in case of the Wired Equivalent Privacy (WEP) protocol in which a checksum function C is used. This function has the homomorphism property over $+$, *i.e.* $C(x + y) = C(x) + C(y)$. Another well-known example, the TMN protocol [TMN89], is detailed in Section 9. The homomorphism property is also crucial in the field of electronic voting protocols [CGS97].

Our results extends previous works [CLT03,LLT05a] for the case of *passive attacks* (the intruder can only listen to messages) that have resulted in a PTIME decision procedure [Del05] for the ACUNh theory. This latter algorithm is an important ingredient to the algorithm for active attacks developed in this paper. Another important ingredient to our algorithm is unification modulo the equational theory ACUNh which has been shown decidable by [GNW00]. Decidability of unification modulo an equational theory E is a necessary condition for decidability of the security property for a bounded number of role instances [CDL05]. Moreover, our algorithm will rely on the fact that unification modulo ACUNh is *finitary*, that is that every problem has a finite set of most general solutions.

The ACUNh equational theory does not fall in any of the results cited above for the active case. Although the equational theory is a combination of the homomorphism axiom with the theory of *exclusive or*, we can not use the combination result by Chevalier and Rusinowitch [CR05] since these two theories are not disjoint. Moreover, their result relies on a model which is different to ours in that it allows to model only a restricted class of protocols. In particular, their model does not cope with the TMN protocol which is described in Section 9. Lastly, the promising approach of H. Comon-Lundh [CL04] which consists in separating the *offline intruder theory* (*i.e.* the capabilities of the intruder to build new messages) from the equational theory will not handle the ACUNh theory since this theory does not enjoy the *finite variant property* [CD05].

Our algorithm is largely inspired by Millen and Shmatikov's approach [MS05] for the equational theories of Abelian groups and Diffie-Hellman exponentiation. Although the overall structure of our proof is the same as theirs there are many differences in the technical details. In particular, we think that our procedures to reduce deducibility constraints to one-step deducibility constraints, and the procedure to solve a special case of quadratic equations in polynomials over the finite field $\mathbb{Z}/2\mathbb{Z}[h]$, remedy some of the shortcomings of [MS05].

2 Preliminaries

2.1 Terms

We use classical notations and terminology on terms, unification and rewrite systems. We write $\mathcal{T}(\mathcal{F}, \mathcal{X})$ for the set of terms built over the finite (ranked) alphabet $\mathcal{F}$ of function symbols and the set of variable symbols $\mathcal{X}$. For our purpose, we assume that $\mathcal{F}$ contains at least the function symbols $\langle ., . \rangle$, $\{.\}_.$, $. + .$ and $h(.)$. The set $\mathcal{T}(\mathcal{F}, \emptyset)$ of ground terms (terms without variables) is also written $\mathcal{T}(\mathcal{F})$. The set of positions of a term t is written $\mathcal{O}(t)$. The subterm of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ at position $p \in \mathcal{O}(t)$ is written $t|_p$. The term obtained by replacing $t|_p$ with s is denoted $t[s]_p$. We refer to any term u that is the same as t everywhere except below p, *i.e.* such that $u[s]_p = t$, as the *linear context* within which the replacement takes place. More precisely, a linear context is a term u with a distinguished position p. The set of variables occurring in t is denoted $\mathit{vars}(t)$.

A *substitution* σ is a mapping from a finite subset of $\mathcal{X}$ called its domain, and written $\mathit{dom}(\sigma)$, to $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Substitutions are extended to endomorphisms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ as usual. We use a postfix notation for their application. The *image* of a substitution σ is the set $\mathit{img}(\sigma) = \{x\sigma \mid x \in \mathit{dom}(\sigma)\}$. Given two terms u and v the *replacement* of u by v, denoted by $[u \mapsto v]$, maps every term t to the term $t[u \mapsto v]$ which is obtained by replacing all occurrences of u in t by v. Note that the result of such replacement is uniquely determined.

2.2 Unification

If E is a set of equations (unordered pairs of terms), we denote by $\mathit{sig}(\mathsf{E})$ the set of all function symbols occurring in E. An E-*context* is a λ-term $\lambda x_1, \ldots, x_n.t$ with $t \in \mathcal{T}(\mathit{sig}(\mathsf{E}), \{x_1, \ldots, x_n\})$, also written $t[x_1, \ldots, x_n]$. The application of a context $t[x_1, \ldots, x_n]$ to arguments $u_1, \ldots, u_n$ is written $t[u_1, \ldots, u_n]$.

We denote by $=_\mathsf{E}$ the least congruence on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $u\sigma =_\mathsf{E} v\sigma$ for all pairs $u = v \in \mathsf{E}$ and substitutions σ. Two terms s, t are E-*unifiable* if there is a substitution σ such that $s\sigma =_\mathsf{E} t\sigma$. Such a substitution is called an E-unifier of s, t. An E-*unification problem* Γ is a finite set of equations between terms. A substitution σ is an E-unifier of Γ if and only if $s\sigma =_\mathsf{E} t\sigma$ holds for each pair (s, t) of terms in Γ. If all terms are in $\mathcal{T}(\mathit{sig}(\mathsf{E}), \mathcal{X})$ the problem is called *elementary* E-unification problem. If terms contains additional "free" constant symbols the problem is called E-unification problem *with constant*, and if terms can contain additional "free" function symbols the problem is called *general* E-unification problem.

Let σ and θ be two E-unifiers, $\sigma \leq \theta$ if and only if there exists a substitution λ such that $x\theta =_\mathsf{E} x\sigma\lambda$ for all variables x; σ is said *more general* than θ. It is a most general E-unifier (E-*mgu*) if for every E-unifier θ, $\sigma \leq \theta$. We say that there is an E-unification algorithm if it is possible, for any unification problem Γ to compute a *complete* set $\sigma_1, \ldots, \sigma_n$ of E-unifiers of Γ. This means that for every E-unifier σ of Γ, there is an index i such that $\sigma_i \leq \sigma$. The *unification type* of E (w.r.t. a signature Σ) is finitary if for every E-unification problem over Σ, the minimal complete set of E-unifiers has a finite cardinality

2.3 Term Rewriting System

A *term rewriting system* (TRS) is a finite set of *rewrite rules* $l \to r$ where $l \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $r \in \mathcal{T}(\mathcal{F}, vars(l))$. Given a TRS $\mathcal{R}$ and a set of equations E, the relation $\to_{\mathcal{R}/\mathsf{E}}$ (*rewriting modulo* E) is defined as follows: $s \to_{\mathcal{R}/\mathsf{E}} t$ if and only if $s =_{\mathsf{E}} u[l\sigma]_p$ and $u[r\sigma]_p =_{\mathsf{E}} t$, for some linear context u, position p in u, rule $l \to r \in \mathcal{R}$, and substitution σ. The rewrite system is $\mathcal{R}/\mathsf{E}$ is *strongly terminating* if there is no infinite chains $t_1 \to_{\mathcal{R}/\mathsf{E}} t_2 \to_{\mathcal{R}/\mathsf{E}} \dots$ and it is *locally confluent* if for every terms t, s_1 and s_2 such that $t \to_{\mathcal{R}/\mathsf{E}} s_1$, $t \to_{\mathcal{R}/\mathsf{E}} s_2$, there exists a term s such that $s_1 \xrightarrow{*}_{\mathcal{R}/\mathsf{E}} s$, $s_2 \xrightarrow{*}_{\mathcal{R}/\mathsf{E}} s$ where $\xrightarrow{*}_{\mathcal{R}/\mathsf{E}}$ is the reflexive and transitive closure of $\to_{\mathcal{R}/\mathsf{E}}$. A rewrite system $\mathcal{R}/\mathsf{E}$ is said to be E-*convergent* if it is both strongly terminating and locally confluent. A term t is in *normal form* (w.r.t. $\to_{\mathcal{R}/\mathsf{E}}$) if there is no term s such that $t \to_{\mathcal{R}/\mathsf{E}} s$. If $t \xrightarrow{*}_{\mathcal{R}/\mathsf{E}} s$ and s is in normal form then we say that s is a normal form of t.

3 Attacker Model

3.1 Inference System

The most widely used deduction relation representing the deduction abilities of an intruder is often referred to as *Dolev-Yao model* [DY81]. Here we augment the intruder abilities by allowing for equational reasoning modulo a given set E of equational axioms. The deduction system is given in Figure 1: the equational theory is taken into account through the normalization function $\downarrow$.

$$\text{Unpairing (UL)}\ \frac{T \vdash \langle u, v\rangle}{T \vdash u} \qquad \text{Compose (C)}\ \frac{T \vdash u_1\ \dots\ T \vdash u_n}{T \vdash f(u_1, \dots, u_n)}\ \text{with } f \in \mathcal{F} \smallsetminus sig(\mathsf{E})$$

$$\text{Unpairing (UR)}\ \frac{T \vdash \langle u, v\rangle}{T \vdash v} \qquad \text{Context}(\mathsf{M_E})\ \frac{T \vdash u_1\ \dots\ T \vdash u_n}{T \vdash C[u_1, \dots, u_n]\downarrow}\ \text{with } C \text{ an } \mathsf{E}\text{-context}$$

$$\text{Decryption (D)}\ \frac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$$

Fig. 1. Dolev-Yao Model Extended with an Equational Theory

The intended meaning of a *sequent* $T \vdash u$ is that the intruder is able to deduce the term $u \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ from the finite set of terms $T \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$. As in the standard Dolev-Yao model, the intruder can compose new terms (C) from known terms, he can also decompose pairs (UL, UR) and decrypt ciphertexts, providing that he can deduce the decryption key (D). Finally, we relax the *perfect cryptography assumption* by allowing the intruder to apply function symbols such as $+$, h. The algebraic properties of these primitives are automatically take into account thanks to the normalization.

In this paper, we are interested in the equational theory $\mathsf{E} = \mathsf{ACUNh}$ which is made up of the well-known axioms of *exclusive or* in combination with an homomorphism symbol. More formally, ACUNh contains the following equations:

- Associativity, Commutativity (AC): $x + (y + z) = (x + y) + z, x + y = y + x$,
- Unit (U): $x + 0 = x$,
- Nilpotence (N): $x + x = 0$,
- Homomorphism (h): $h(x + y) = h(x) + h(y)$.

Given an homomorphic symbol h, the notation $h^n(t)$ represents the term t if $n = 0$ and $h(h^{n-1}(t))$ otherwise. We represent the ACUNh equational theory by an AC-convergent rewrite system. This can be obtained by orienting from left to right the equations (U), (N), (h) and by adding the consequence $h(0) \to 0$ (see [LLT05a] for details). After each deduction step, the term u obtained is reduced to its normal form $u\downarrow$. Equivalence modulo AC is easy to decide, so we omit the equality rule for AC and just work with equivalence classes modulo AC. More generally, along this paper, we consider implicitly that terms are kept in normal forms, hence we write u (resp. $u\sigma$) instead of $u\downarrow$ (resp. $u\sigma\downarrow$).

This deductive system is equivalent in deductive power to a variant of the system in which terms are not automatically normalized, but in which arbitrary equational proofs are allowed at any moment of the deduction (see [Del05,LLT05a]). The inference system described in Figure 1 deals with symmetric encryption only. However, it is not difficult to design a similar deduction system for asymmetric encryption and to extend the result of this paper to this new inference system.

3.2 Factors, Subterms

A term t is *standard* if and only if it is not of the form $f(u)$ for some term u and some $f \in sig(E)$.

Definition 1. *Let t be a term in normal form. We have $t = C[t_1, \dots, t_n]$ for some standard terms $t_1, \dots, t_n$ and an E-context C. The set $Fact_{\mathsf{E}}(t)$ of* factors *of t is defined by $Fact_{\mathsf{E}}(t) = \{t_1, \dots, t_n\}$. The set $St_{\mathsf{E}}(t)$ of* subterms *of t is the smallest set such that:*

- *$t \in St_{\mathsf{E}}(t)$,*
- *if $f(t_1, \dots, t_n) \in St_{\mathsf{E}}(t)$ is standard then $t_1, \dots t_n \in St_{\mathsf{E}}(t)$,*
- *if $s \in St_{\mathsf{E}}(t)$ is not standard then $Fact_{\mathsf{E}}(s) \subseteq St_{\mathsf{E}}(t)$.*

These notations are extended as expected to sets of terms: $Fact_{\mathsf{E}}(T)$ (resp. $St_{\mathsf{E}}(T)$) is the union of the sets $Fact_{\mathsf{E}}(t)$ (resp. $St_{\mathsf{E}}(t)$) for every term t occurring in T. Note that, by definition, 0 is not a standard term and the factors of any term are necessarily standard.

Example 1. Let $t_1 = h^2(a)+b+c$, $t_2 = h(\langle a, b\rangle)+c$ and $t_3 = \langle a+b+c, d\rangle$. We have $Fact_{\mathsf{E}}(t_1) = \{a, b, c\}$, $St_{\mathsf{E}}(t_1) = \{t_1, a, b, c\}$, $Fact_{\mathsf{E}}(t_2) = \{\langle a, b\rangle, c\}$, and $St_{\mathsf{E}}(t_2) = \{t_2, \langle a, b\rangle, a, b, c\}$, $Fact_{\mathsf{E}}(t_3) = \{t_3\}$, and $St_{\mathsf{E}}(t_3) = \{t_3, a + b + c, d, a, b, c\}$.

Definition 2. *A substitution σ is* non-collapsing *w.r.t. a set T of terms iff:*

$$\forall u, v \in St_{\mathsf{E}}(T) \setminus \mathcal{X} : u\sigma =_{\mathsf{E}} v\sigma \Rightarrow u =_{\mathsf{E}} v$$

Definition 3. *The set of* non-standard subterms $NSt(t)$ *of a term t is*

$$\begin{array}{lll} NSt(f(t_1, \dots, t_n)) = \bigcup_{i=1}^{n} NSt(t_i) & & \textit{if } f \notin sig(\mathsf{E}) \\ NSt(t) & = \{t\} \cup \bigcup_{s \in Fact_E(t) \setminus \mathcal{X}} NSt(s) & \textit{otherwise} \end{array}$$

Example 2. Let $t = h(X_1)+X_2+\langle X_3, X_4+X_5\rangle$, then $NSt(t) = \{t, X_3, X_4+X_5\}$.

3.3 Proofs

Definition 4. *A* proof P *of $T \vdash u$ is a finite tree such that*

- *every leaf of P labeled with $T \vdash v$ is such that $v \in T$,*
- *for every node of P labeled with $T \vdash v$ having n sons ($n \geq 1$) labeled with $T \vdash v_1, \dots, T \vdash v_n$, there is an instance $\dfrac{T \vdash v_1 \ \ \dots \ \ T \vdash v_n}{T \vdash v}$ (R) of an inference rule in Figure 1. If this node labeled with $T \vdash v$ is the root of P, we say that* P ends *with an instance of* (R).
- *the root of P is labeled with $T \vdash u$.*

Note that the terms in the proof are not necessarily ground. The *size* of a proof P, denoted by $|P|$, is the number of nodes in P. A proof P of $T \vdash u$ is *minimal* if there is no proof P' of $T \vdash u$ such that $|P'| < |P|$. A proof P of $T \vdash u$ is a *decomposition proof* in any of the following cases:

- $|P| = 1$ (this means that P is reduced to a leaf),
- P ends with an instance of a decomposition rule (*i.e.* $(\mathsf{UL}, \mathsf{UR}, \mathsf{D})$),
- P ends with an instance of $(\mathsf{M}_{\mathsf{E}})$ and u is a standard term.

Definition 5. *A term u is* R-one-step deducible *from a set of terms T in any of the following cases:*

- *$T \vdash u$ is a proof of $T \vdash u$ (i.e, $u \in T$),*
- *there exists $u_1, \dots, u_n$ such that $\dfrac{T \vdash u_1 \ \ \dots \ \ T \vdash u_n}{T \vdash u}$ (R) is a proof of $T \vdash u$.*

We say that u is one-step deducible *from T if there exists an inference rule (R) in Figure 1 such that u is R-one-step deducible from T. We say also that u is* DY-one-step deducible *from T if* $\mathsf{R} \in \{\mathsf{C}, \mathsf{UL}, \mathsf{UR}, \mathsf{D}\}$.

Example 3. Consider the ACUNh theory, let $T = \{a+h(a), b\}$, the proof P below is a proof of $T \vdash a + h^3(a) + h(b)$. It is made up of an instance of the rule $(\mathsf{M}_{\mathsf{E}})$ with $C = x_1 + h(x_1) + h^2(x_1) + h(x_2)$.

$$\frac{T \vdash a + h(a) \ \ T \vdash b}{T \vdash a + h^3(a) + h(b)} \, (\mathsf{M}_{\mathsf{E}})$$

Since $a + h^3(a) + h(b)$ is not standard, P is not a decomposition proof. We have $|P| = 3$ and $a + h^3(a) + h(b)$ is M_{E}-one-step deducible from T but is not DY-one-step deducible from T since $\mathsf{M}_{\mathsf{E}} \notin \{\mathsf{C}, \mathsf{UL}, \mathsf{UR}, \mathsf{D}\}$.

3.4 Locality Lemma

Now, we can state the following locality lemma. This notion was first introduced by McAllester [McA93] in order to characterize theories with a deduction problem decidable in PTIME. This result, proven in [Del05], allows us to focus on proof trees that involve only some particular terms.

Lemma 1. *A minimal proof P of $T \vdash u$ contains only terms in $St_{\mathsf{E}}(T \cup \{u\})$. Moreover, if P is a decomposition proof, then P contains only terms in $St_{\mathsf{E}}(T)$.*

4 Constraint System

4.1 Well-Defined Constraint System

Definition 6. *A* constraint *(resp. one-step constraint, $\mathsf{M_E}$ constraint) is a sequent of the form $T \Vdash u$ (resp. $T \Vdash_1 u$, $T \Vdash_{\mathsf{M_E}} u$) where T is a finite subset of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ and $u \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. We call T (resp. u) the* hypothesis set *(resp. the* target*) of the constraint. A* system of constraints *is a sequence of constraints. A solution to a system $\mathcal{C}$ of constraints is a substitution σ such that:*

- *for every $T \Vdash u \in \mathcal{C}$ there exists a proof of $T\sigma \vdash u\sigma$*
- *for every $T \Vdash_1 u \in \mathcal{C}$ $u\sigma$ is one-step deducible from $T\sigma$*
- *for every $T \Vdash_{\mathsf{M_E}} u \in \mathcal{C}$, $u\sigma$ is $\mathsf{M_E}$-one-step deducible from $T\sigma$.*

If $\mathcal{F}'$ is a sub-signature of $\mathcal{F}$ then a solution σ to a constraint system is called a $\mathcal{F}'$-solution if $x\sigma \in \mathcal{T}(\mathcal{F}', \mathcal{X})$ for every $x \in dom(\sigma)$.

Note that, if σ is solution to a constraint (resp. one-step constraint, $\mathsf{M_E}$ constraint) c, then $\sigma\theta$ is also a solution to c for every substitution θ.

Definition 7. *A constraint system $\mathcal{C} = \{T_i \Vdash u_i\}_{1 \leq i \leq k}$ is* well-formed *if:*

1. (monotonicity) *$0 \in T_0$ and for all $i < k$, $T_i \subseteq T_{i+1}$,*
2. (origination) *for all $i \leq k$, for all $x \in vars(T_i)$, there exists $j < i$ such that $x \in vars(u_j)$.*

We say that $\mathcal{C}$ is well-defined *if for every substitution θ, $\mathcal{C}\theta$ is well-formed.*

This notion of well-definedness is defined in a similar way on systems of one-step (resp. $\mathsf{M_E}$) constraints. This definition is due to Millen and Shmatikov. In [MS05] they show that "reasonable" protocols, in which legitimate protocol participants only execute deterministic steps (up to the generation of random nonces) always lead to a well-defined constraint system. In the following we will only consider well-defined protocols. This allows us to restrict our attention to well-defined constraint systems.

4.2 Conservative Solutions

The completeness of our decision procedure is ensured by the existence of a *conservative* solution (Lemma 2) which does not introduce any new structure. Moreover, conservative solutions allow us to lift the locality Lemma (see Lemma 3). The proofs of Lemmas 2 and 3 are detailed in Appendix A.

Definition 8. *Let $\mathcal{C}$ be a constraint system and σ a substitution, σ is* conservative *w.r.t. $\mathcal{C}$ if and only if for all $x \in vars(\mathcal{C})$, $Fact_{\mathsf{E}}(x\sigma) \subseteq (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma$.*

Lemma 2. *Let $\mathcal{C}$ be a well-defined constraint system. If there exists a solution σ to $\mathcal{C}$ then there exists a conservative one.*

Example 4. Consider the following well-defined constraint system $\mathcal{C}$:

$$\begin{array}{l} 0, a, h(b) \quad \Vdash h(x) \\ 0, a, h(b), x \Vdash \langle a, b \rangle \end{array}$$

One solution is $\sigma = \{x \mapsto \langle a, a \rangle + b\}$. This solution is not conservative w.r.t. $\mathcal{C}$. Indeed $Fact_{\mathsf{E}}(\langle a, a \rangle + b) = \{\langle a, a \rangle, b\}$, and $\langle a, a \rangle$ does not belong to $(St_{\mathsf{E}}(\mathcal{C}) \setminus \{x\})\sigma$ which is equal to $\{0, h(b), b, h(\langle a, a \rangle + b), \langle a, b \rangle, a\}$. However, as it is said in Lemma 2, there is a conservative solution: $\{x \mapsto b\}$.

Lemma 3. *Given a conservative solution σ of $\mathcal{C} = \{C_1, \ldots, C_k\}$. For each $i \leq k$, there exists a proof P_i of $C_i\sigma$ which involves only terms in $St_{\mathsf{E}}(\mathcal{C})\sigma$.*

5 Some Results about E-Unification

The following theorem (proved in Appendix B) states that unification in ACUNh is finitary. This is an important ingredient of our decision procedure (see Lemma 6).

Theorem 1. *Unification in the theory ACUNh is finitary and there exists an algorithm to compute a complete finite set of unifiers of any unification problem.*

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 is denoted by . and 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 $(h^2+1) \odot (X+a) = h^2(X) + X + h^2(a) + a$. Conversely a term t such that $vars(t) = \{X_1, \ldots, X_p\}$ can be written $t^{X_1} \odot X_1 + \ldots + t^{X_p} \odot X_p + t_0$ for some $t^{X_1}, \ldots, t^{X_p} \in \mathbb{Z}/2\mathbb{Z}[h]$, and t_0 a ground term.

The following technical lemma will be used in the proof of Lemma 5.

Lemma 4. *Let P be a unification problem in the theory $\mathsf{E} = \mathit{ACUNh}$ (including free function symbols) and θ be an mgu_{E} of P. Then for all $x \in dom(\theta)$ and $v \in St_{\mathsf{E}}(x\theta) \setminus vars(x\theta)$ there exists $t \in St_{\mathsf{E}}(P)$ such that $v =_{\mathsf{E}} t\theta$.*

To a polynomial $P(h) = \sum_{i=0}^{i=n} b_i h^i$ with $b_i \in \{0,1\}$, we associate the number $nb(P)$ whose representation in base 2 is $b_n \dots b_0$. This correspondence is one-one. We define a total ordering $<$ on polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$ by $P < P'$ if and only if $nb(P) < nb(P')$. This is a total Noetherian ordering (no infinite decreasing sequence exists).

6 From Constraints to M_{E} Constraints

To reduce the satisfiability of a constraint system to the satisfiability of M_{E} constraint system, we proceed in two steps:

1. Firstly, we reduce our problem to the satisfiability of one-step (but not necessary M_{E}) constraints (Lemma 5).
2. Secondly, we reduce the satisfiability of one-step constraint system to the satisfiability of M_{E} constraint system (Lemma 6).

Proofs are given in Appendix C.

```
Input: C = {T_1 ⊩ u_1, ..., T_k ⊩ u_k}
Output: C'
Algorithm:
  guess S ⊆ St_E(C).
  for all s ∈ S, guess j(s) ∈ {1, ..., k}.
  C' := ∅
  for i = 1 to k do
     let S_i := {s | j(s) = i}.
     choose a total ordering on S_i (S_i = {s_i^1, ..., s_i^{ki}})
     for j = 1 to ki do
        T := T_i ∪ S_1 ... ∪ S_{i-1} ∪ {s_i^1,..., s_i^{j-1}}
        C' := C' ∪ {T ⊩_1 s_i^j}
     end
     C' := C' ∪ {T ⊩_1 u_i}
  end
  return C'.
```

Fig. 2. From Constraints to One-Step Constraints

The non-deterministic algorithm described in Figure 2 allows us to reduce the satisfiability of a system of constraints to the satisfiability of a system of one-step constraints. First, it guesses among the subterms of $\mathcal{C}$ those which are going to be deduced by the intruder and inserts each deducible subterm in the constraint system. The completeness of this step of the procedure is essentially due to the existence of a conservative solution (Lemma 2) and the lifting locality lemma (Lemma 3). In the resulting constraint system, every constraint can be solved by application of a single inference rule.

Lemma 5. *Let $\mathcal{C}$ be a well-defined system of constraints. Let $\mathscr{C}'$ be the set of constraint systems obtained by applying the algorithm of Figure 2 on $\mathcal{C}$.*

1. *$\mathscr{C}'$ is a finite set of well-defined system of one-step constraints.*
2. *Let $\mathcal{C}' \in \mathscr{C}'$. If σ is a solution to $\mathcal{C}'$ then σ is a solution to $\mathcal{C}$.*
3. *If σ is a conservative solution to $\mathcal{C}$ then there exists $\mathcal{C}' \in \mathscr{C}'$ such that σ is a solution to $\mathcal{C}'$.*
4. *For any $\mathcal{C}' \in \mathscr{C}'$, σ is conservative w.r.t. $\mathcal{C}$ if and only if σ is conservative w.r.t. $\mathcal{C}'$.*

Lemma 6 allows to reduce the satisfiability of a system of one-step constraints to the satisfiability of a system of $\mathsf{M_E}$ constraints. We first guess a set of equalities between subterms and thus obtain a unification problem which has a finite and complete set of solutions thanks to Theorem 1. We apply the unifier to the constraint system. Then, one-step constraints which can be solved by the application of a standard inference rule, *i.e.* (D, UL, UR, C), can be determined by syntactic inspection. Hence, we can eliminate all constraints which can be satisfied by a single application of an inference rule other than ($\mathsf{M_E}$).

Lemma 6. *Given $\mathcal{C}$ a well-defined system of one-step constraints. Let $\mathcal{P} = \{\bigwedge_{(s_1,s_2)\in S'} s_1 = s_2 \mid S' \subseteq St_E(\mathcal{C})^2\}$. Let $R \in \mathcal{P}$ and $\theta \in mgu_{\mathsf{E}}(R)$. Let $\mathcal{C}_\theta = \{T\theta \Vdash_{\mathsf{M_E}} u\theta \mid T \Vdash_1 u \in \mathcal{C}$ and $u\theta$ is not DY-one-step deducible from $T\theta\}$.*

1. *There are only finitely many outputs for a given input $\mathcal{C}$. Each of them is a well-defined system of $\mathsf{M_E}$ constraints.*
2. *If there exists $\mathcal{C}_\theta$ (obtained by the procedure above) which has a solution then $\mathcal{C}$ has a solution.*
3. *If $\mathcal{C}$ has a conservative solution then there exists $\mathcal{C}_\theta$ (obtained by the procedure above) which has a non-collapsing solution.*

7 Dealing with $\mathsf{M_E}$ Constraints

Now, we have to solve well-defined $\mathsf{M_E}$ constraint systems. In the remainder, we consider a $\mathsf{M_E}$ constraint system $\mathcal{C}$ of the following form:

$$\begin{array}{r}
t_1, \ldots, t_n \Vdash_{\mathsf{M_E}} u_1 \\
t_1, \ldots, t_n, t_{n+1} \Vdash_{\mathsf{M_E}} u_2 \\
\vdots \\
t_1, \ldots, t_n, t_{n+1}, \ldots, t_{n+k-1} \Vdash_{\mathsf{M_E}} u_k
\end{array}$$

We assume (w.l.o.g) that the hypotheses (*i.e.* $t_1, \ldots t_{n+i}$) of the $i+1^{th}$ constraint contain exactly one term more than the hypotheses of the i^{th} constraint. This can be achieved by duplicating some terms or by adding some constraints.

The proofs of this section can be found in Appendix D.

7.1 Factor Preservation

Definition 9. *A constraint system is* factor-preserving *if for all i, $1 \leq i \leq k$, we have that $Fact_E(u_i) \setminus \mathcal{X} \subseteq \bigcup_{j=1}^{j=n+i-1} Fact_E(t_j)$.*

Lemma 7. *If a well-defined $\mathsf{M_E}$-constraint system $\mathcal{C}$ has a non-collapsing solution then it is factor-preserving.*

7.2 Another Characterization of Well-Definedness

Let $vars(\mathcal{C}) = \{X_1, \ldots, X_p\}$. As we have already seen in Section 5, every t_i (or u_i) can be written $t_i^{X_1} \odot X_1 + \ldots t_i^{X_p} \odot X_p + t_i^0$ with $t_i^{X_v}$ in $\mathbb{Z}/2\mathbb{Z}[h]$ and t_i^0 is a sum of standard terms. We will denote with $\boldsymbol{t_i}$ the vector $(t_i^{X_1}, \ldots, t_i^{X_p})$.

Definition 10. *Let $\mathcal{V} = \{v_1, \ldots, v_m\}$ be a subset of $\mathbb{Z}/2\mathbb{Z}[h]^n$. $\mathcal{V}$ is* independent *if whenever there exist $\alpha_i \in \mathbb{Z}/2\mathbb{Z}[h]$ such that $\alpha_1 v_1 + \ldots + \alpha_m v_m = 0$ then $\alpha_i = 0$ for all $1 \leq i \leq m$. Otherwise $\mathcal{V}$ is* dependent.

Proposition 1. *Let A be a matrix $n \times m$ over $\mathbb{Z}/2\mathbb{Z}[h]$ such that the n row vectors are independent ($n \leq m$) then:*

$$\exists Q \in \mathbb{Z}/2\mathbb{Z}[h], \forall b \in \mathbb{Z}/2\mathbb{Z}[h]^n, \exists X \in \mathbb{Z}/2\mathbb{Z}[h]^m \;\; A \cdot X = Q \cdot b \tag{1}$$

Moreover, a coefficient Q is computable as a determinant of a sub matrix of A.

Given $\mathcal{C}$ a well-defined $\mathsf{M_E}$ constraint system, we construct first the set L containing the indexes of the *defining constraints* by using the following algorithm:

```
Input: C = {T_1 ⊩_ME u_1, ..., T_k ⊩_ME u_k}
Output: L
Algorithm:
  L := ∅;
  for i = 1 to k do
    if {u_i} ∪ {u_j | j ∈ L } is independent then L := L ∪ {i};
end
return L.
```

Let $\mathcal{B}_i = \{\boldsymbol{u_j} \mid j \in L, j \leq i\}$, and $\mathcal{B} = \mathcal{B}_n$. By construction of L, each $\mathcal{B}_i$, and hence $\mathcal{B}$, is independent. Let Q_{max} be a coefficient Q witnessing equation (1) for the matrix $\mathcal{B}$.

Example 5. (running example) To illustrate our procedure, we consider the following well-defined $\mathsf{M_E}$ constraint system:

$$
\begin{array}{ll}
0, h(a)+a, b+h^2(a) & \Vdash_{\mathsf{M_E}} h(X_1)+h^2(X_2) \\
0, h(a)+a, b+h^2(a), X_1+h(X_2) & \Vdash_{\mathsf{M_E}} X_1+a \\
0, h(a)+a, b+h^2(a), X_1+h(X_2), h(X_1)+h(a) & \Vdash_{\mathsf{M_E}} h(X_1)+h^2(X_2)+X_1+a
\end{array}
$$

We have $\boldsymbol{u_1} = (h, h^2)$, $\boldsymbol{u_2} = (1, 0)$ and $\boldsymbol{u_3} = (1+h, h^2)$. The algorithm returns $L = \{1, 2\}$ and we obtain $Q_{max} = det(\boldsymbol{u}_1, \boldsymbol{u}_2) = h^2$.

Lemma 8. *Let $\mathcal{C} = \{t_1, \ldots, t_{n+i-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ be a well-defined and factor-preserving $\mathsf{M_E}$ constraint system. Then for every $i \leq k$ and $s \in NSt(t_{n+i-1})$ the set $\{\boldsymbol{s}\} \cup \mathcal{B}_{i-1}$ is dependent.*

Example 6. Consider the following constraint system :

$$
\begin{array}{r}
0, a \Vdash_{\mathsf{M_E}} X_1 + X_2 \\
0, a, b \Vdash_{\mathsf{M_E}} X_1 \\
0, a, b, \langle h(X_1), a\rangle + \langle h(X_2), a\rangle \Vdash_{\mathsf{M_E}} a + b
\end{array}
$$

This system is well-defined and factor-preserving. We have $L = \{1, 2\}$, $\boldsymbol{u_1} = (1, 1)$ and $\boldsymbol{u_2} = (1, 0)$. We have $NSt(\langle h(X_1), a\rangle + \langle h(X_2), a\rangle) = \{\langle h(X_1), a\rangle + \langle h(X_2), a\rangle; h(X_1); h(X_2)\}$. The sets $\{(1, 1), (1, 0), (0, 0)\}$, $\{(1, 1), (1, 0), (h, 0)\}$ and $\{(1, 1), (1, 0), (0, h)\}$ are dependent.

If we omit the second constraint, we obtain $L = \{1\}$ and for instance the sets $\{(1, 1), (h, 0)\}$ is independent. However, note that such a constraint system is not well-defined as witnessed by the substitution $\theta\colon X_i \mapsto X_i + W$.

7.3 Reducing the Signature

We will show in Lemma 9 that we can reduce the satisfiability of $\mathsf{M_E}$ constraint systems to the satisfiability of $\mathsf{M_E}$ constraint systems over a signature consisting only of $+$, h, and a set of constants.

If $\rho : M \to N$ is a replacement, that is a bijection between two finite sets of terms M and N, then we denote for any term t by t^ρ the term obtained by replacing in t any top-most occurrence of a subterm $s \in M$ by $s\rho$. This extends in a natural way to constraint systems, and to substitutions by setting $x(\sigma^\rho) = (x\sigma)^\rho$ for all variables $x \in dom(\sigma)$.

Lemma 9. *Let $\mathcal{C}$ be a well-defined factor-preserving $\mathsf{M_E}$ constraint system and $F = Fact_E(\mathcal{C}) \backslash \mathcal{X}$. Let $\mathcal{F}_0$ be a set of new constant symbols of the same cardinality as F and $\rho : F \to \mathcal{F}_0$ a bijection.*

1. *$\mathcal{C}^\rho$ is well-defined.*
2. *$vars(\mathcal{C}^\rho) = vars(\mathcal{C})$.*
3. *If $\mathcal{C}$ has a non-collapsing solution then $\mathcal{C}^\rho$ has a $\mathcal{F}_0 \cup \{h, +\}$-solution.*
4. *If $\mathcal{C}^\rho$ has a $\mathcal{F}_0 \cup \{h, +\}$-solution then $\mathcal{C}$ has a solution.*

7.4 Solving $\mathsf{M_E}$ Constraint Systems over $\{h, +\} \cup \mathcal{F}_0$

We may by Lemma 7 assume that we have a factor-preserving $\mathsf{M_E}$ constraint system. By Lemma 9 satisfiability of such a system can be reduced to satisfiability of a $\mathsf{M_E}$ constraint system over a signature $\{h,+\}\cup\mathcal{F}_0$ where $\mathcal{F}_0$ is a finite set of constants. Satisfiability of such an $\mathsf{M_E}$ constraint system $\mathcal{C}$ is equivalent to the satisfiability of the following system $\mathcal{S}$ of equations between terms.

$$
\begin{array}{l}
z[1,1] \odot t_1 + \ldots + z[1,n] \odot t_n = u_1 \\
z[2,1] \odot t_1 + \ldots + z[2,n] \odot t_n + z[2,n+1] \odot t_{n+1} = u_2 \\
\vdots \\
z[p,1] \odot t_1 + \ldots + z[p,n] \odot t_n + \ldots + z[p,n+p-1] \odot t_{n+k-1} = u_k
\end{array}
$$

The variables $z[i,j]$, called *context variables*, take their value in $\mathbb{Z}/2\mathbb{Z}[h]$. Let $\mathcal{Z} = \{z[i,j] \mid 1 \le i \le k, 1 \le j \le n+i-1\}$. The u_i's and t_i's are terms that are not necessarily ground.

Example 7. (running example) For sake of clarity we omit the contexts associated to 0 and we note $t_1 = h(a) + a$ and $t_2 = b + h^2(a)$.

$$
\begin{array}{l}
z[1,1] \odot t_1 + z[1,2] \odot t_2 = h(X_1) + h^2(X_2) \\
z[2,1] \odot t_1 + z[2,2] \odot t_2 + z[2,3] \odot (X_1 + h(X_2)) = X_1 + a \\
z[3,1] \odot t_1 + z[3,2] \odot t_2 + z[3,3] \odot (X_1 + h(X_2)) + z[3,4] \odot (h(X_1) + h(a)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = h(X_1) + h^2(X_2) + X_1 + a
\end{array}
$$

Definition 11. *Let $\mathcal{C}$ be a well-defined $\mathsf{M_E}$ constraint system over the signature $\{h,+\}\cup\mathcal{F}_0$ and $\mathcal{S}(\mathcal{C})$ be the system of equations obtained from $\mathcal{C}$. A solution to $\mathcal{S}(\mathcal{C})$ is a couple $(\rho : \mathcal{Z} \mapsto \mathbb{Z}/2\mathbb{Z}[h], \theta : vars(\mathcal{C}) \mapsto \mathcal{T}(\{h,+\}\cup\mathcal{F}_0))$ such that all the equations of $\mathcal{S}(\mathcal{C})\rho\theta$ are satisfied.*

We split the *context variables* $\mathcal{Z}$ in two parts, those which stem from L and the others. More formally, $\mathcal{Z}_L = \{z[i,j] \mid i \in L \text{ and } 1 \le j < n+i\}$.Note that the variables of $\mathcal{Z}$ are totally ordered by the lexicographic order of the indices of variables, that is $z[i,j] \prec z[i',j']$ iff $i < i'$, or else $i = i'$ and $j < j'$.

The following Lemma is the crucial point in the proof of Lemma 11.

Lemma 10. *Let $\mathcal{S}(\mathcal{C})$ be a system of equations obtained from a well-defined $\mathsf{M_E}$ constraint system $\mathcal{C}$ over the signature $\{h,+\}\cup\mathcal{F}_0$. If $\mathcal{S}(\mathcal{C})$ has a solution then there exists σ a solution to $\mathcal{S}(\mathcal{C})$ such that for all $z \in \mathcal{Z}_L$, $0 \le z\sigma < Q_{max}$.*

Lemma 11. *Given $\mathcal{C}$ a well-defined $\mathsf{M_E}$constraint system. It is decidable whether $\mathcal{S}(\mathcal{C})$ has a solution.*

Example 8. (running example) Thanks to Lemma 10, we know that $z[1,1]$, $z[1,2]$, $z[2,1]$, $z[2,2]$ and $z[2,3]$ are bounded by h^2, the value of Q_{max}. We choose

$\rho_1 = \{z[1,1] \mapsto 0; z[1,2] \mapsto h; z[2,1] \mapsto h+1; z[2,2] \mapsto 1; z[2,3] \mapsto 0\}$. We do the replacement on the two first equations:

$$\begin{array}{l} h \odot (b + h^2(a)) = h(X_1) + h^2(X_2) \\ (h+1) \odot (h(a) + a) + 1 \odot (b + h^2(a)) = X_1 + a \end{array}$$

This completely determines the value of X_1 and X_2: $\theta = \{X_1 \mapsto b, X_2 \mapsto h(a)\}$. Lastly, we can apply the substitution θ on the third equation to obtain:

$$\begin{aligned} z[3,1] \odot (h(a)+a) + z[3,2] \odot (b + h^2(a)) + z[3,3] \odot (b + h^2(a)) + \\ z[3,4] \odot (h(b) + h(a)) = h(b) + h^3(a) + b + a \end{aligned}$$

We can decide whether there exists a solution by solving the following system of linear equations over $\mathbb{Z}/2\mathbb{Z}[h]$. Notice that $u_3 = u_1 + u_2$, hence there is a solution $\rho_2 = \{z[3,1] \mapsto h+1; z[3,2] \mapsto h+1; z[3,3] \mapsto 0; z[3,4] \mapsto 0\}$ and $(\rho_1 \cup \rho_2, \theta)$ is a solution to the system of equations described in Example 7.

8 Main Result

Theorem 2. *Satisfiability of a well-defined constraint system $\mathcal{C}$ is decidable in presence of the equational theory* ACUNh.

Proof. The procedure described along the paper is sound and complete. Let $\mathcal{C}$ be a well-defined constraint system.

Soundness: Let $\mathcal{C}_1$ be some factor-preserving $\mathsf{M_E}$ constraint system obtained by applying the first part of our procedure on $\mathcal{C}$. Thanks to Lemma 5 and 6, $\mathcal{C}_1$ is well-defined since $\mathcal{C}$ is well-defined. Let $\mathcal{C}_2$ be the constraint system obtained from $\mathcal{C}_1$ by replacing all factors by different constants. $\mathcal{C}_2$ is well-defined thanks to Lemma 9. Assume that $\mathcal{S}(\mathcal{C}_2)$ (the system of equations associated to $\mathcal{C}_2$) has a solution. We easily deduce that $\mathcal{C}_2$ has a solution, hence by Lemma 9 that $\mathcal{C}_1$ has a solution, and by Lemma 5 and 6 that $\mathcal{C}$ has a solution.

Completeness: Assume that σ is a solution to $\mathcal{C}$. Thanks to Lemma 2, we can assume that σ is conservative w.r.t. $\mathcal{C}$. Let $\mathscr{C}'$ be the finite set of well-defined one-step constraint systems obtained by applying the algorithm described in Figure 2 on $\mathcal{C}$. By Lemma 5, we know that there exists $\mathcal{C}' \in \mathscr{C}'$ such that σ is a conservative solution of $\mathcal{C}'$. By Lemma 6, we know that there exists $\mathcal{C}_\theta$ a well-defined $\mathsf{M_E}$ constraint system which has a non-collapsing solution. Hence, $\mathcal{C}_\theta$ is factor-preserving due to Lemma 7. By Lemma 9, $\mathcal{C}_\theta^\rho$ has solution over $\{+, h\} \cup \mathcal{F}_0$ Then, Lemma 11 allows us to conclude. $\square$

9 Example: TMN's Protocol

We illustrate our constraint solving technique by applying it to a protocol due to Tatebayashi, Matsuzaki and Newman (TMN) [TMN89].

9.1 Description

The TMN protocol is a symmetric key distribution protocol involving three participants and four exchanges of messages[4]. The only trusted key is the public key of the server S.

$$\begin{array}{ll}
1. & A \to S : A, B, \{K_a\}_{\mathsf{pub}(S)} \\
2. & S \to B : A \\
3. & B \to S : B, A, \{K_b\}_{\mathsf{pub}(S)} \\
4. & S \to A : B, K_b \oplus K_a
\end{array}$$

When the agent A wants to communicate with B, he chooses a key K_a, encrypts it, and sends it to the server S (msg 1). The server sends a message to B informing that A wants to communicate with him (msg 2). B chooses a key K_b, encrypts it, and sends it to S (msg 3). The server forms the Vernam encryption of the two keys K_a and K_b, and returns this message to A (msg 4). Now, knowing $K_a \oplus K_b$ and K_a, A can retrieve the session key K_b. To prevent replay attacks, the server S verifies that the keys K_a and K_b have not been used in previous sessions.

The protocol employs two kind of encryptions: the standard asymmetric encryption and the Vernam encryption. In the first encryption scheme, everyone who knows m can produce $\{m\}_{\mathsf{pub}(S)}$, but only the server knows how to decrypt such a ciphertext. This can be achieved by using RSA encryption. The Vernam encryption, denoted $\oplus$, simply consists in applying the *exclusive or* operator on the message and the key.

In [TMN89], an attack on the protocol, due to Simmons, was described. The attack makes use of the fact that if the asymmetric encryption is implemented using RSA, then we have the following homomorphic property:

$$\{x \times y\}_{\mathsf{pub}(S)} = \{x\}_{\mathsf{pub}(S)} \times \{y\}_{\mathsf{pub}(S)}$$

where $\times$ represents multiplication modulo the public modulus.

The attack presented in [Sim94] is based on the fact that the intruder can eavesdrop the message $\{K_b\}_{\mathsf{pub}(S)}$ exchanged between A and B during a normal session of the protocol. The intruder I can fool the server during a new session to obtain K_b. Firstly, I sends $\{K_i\}_{\mathsf{pub}(S)}$, then he sends $\{K_i\}_{\mathsf{pub}(S)} \times \{K_b\}_{\mathsf{pub}(S)}$ which is equal to $\{K_i \times K_b\}_{\mathsf{pub}(S)}$ thanks to the homomorphism property. The server thinks that it is a fresh key and plays the rest of the protocol. In particular, he sends $(K_i \times K_b) \oplus K_i$ to I. The intruder, knowing K_i can retrieve K_b, the session key established during a previous session between two honest agents A and B. Note that the simple attack consisting in replaying $\{K_b\}_{\mathsf{pub}(S)}$ at step 1 does not work since it is assumed that S can detect keys that have already been used in previous sessions.

[4] For sake of notations, we often omit the pairing function symbol $\langle \_, \_ \rangle$.

9.2 Modeling

In the remainder, we are going to use the homomorphism function symbol h to model RSA encryption with the public key of S. The fact that the decryption key of the server S is a trusted key is take into account by the fact that the intruder can not obtain m from $h(m)$. In our model, we can not model the fact that the homomorphism is over a multiplication, *i.e.* an Abelian group operator. It is for this that we choose to model it by an homomorphism over $\oplus$. This operator satisfies more algebraic properties, hence this abstraction could find some non realistic attacks. Lastly, in the constraint solving model, we can not deal with the test perform by the server to detect keys that have already been used in previous sessions. Hence, we simply omit it. For sake of clarity, we restrict the number of constraints by considering the minimum needed to retrieve the attack due to Simmons.

During the first session, the attacker eavesdrops the communication network to increase his initial knowledge $T_0 = \{A, B, C, S, I, h(K_b), h(K_a), K_i\}$. The intruder knows the identity of the agents, the two ciphertexts he has seen on the network during the execution of the first session and the key K_i that he shares with his accomplice. Now, we are going to write the constraint system corresponding to the execution of the session in which I sends his two requests to the server S. The constraint system obtained is the following one:

$$\mathcal{C} := \begin{cases} T_0 \Vdash X_a, X_b, h(Y_a) & S \text{ receives a request from } I \\ T_1 := T_0, S, X_a & S \text{ answers the request to } X_b \\ T_1 \Vdash X_b, X_a, h(Y_b) & S \text{ receives the answer of } X_b \\ T_2 := T_1, S, X_b, Y_a \oplus Y_b & S \text{ answers the request to } I \\ T_2 \Vdash K_b & \text{the secret } K_b \text{ is revealed} \end{cases}$$

9.3 Constraint Solving

First, we guess a subset of $St_E(\mathcal{C})$, we split this set in three subsets S_0, S_1 and S_2 and we order the terms in each subset. We have $S_0 = \{X_a, X_b, h(Y_a)\}$ $S_1 = \{Y_b, h(Y_b)\}$ and $S_2 = \{K_b\}$. The non-deterministic algorithm of Figure 2 apply on $\mathcal{C}$ with the choices described above, returns the following set of one-step constraints.

$$\mathcal{C}' := \begin{cases} T_0 \Vdash_1 X_a & T_1, S_0 \Vdash_1 Y_b \\ T_0, X_a \Vdash_1 X_b & T_1, S_0, Y_b \Vdash_1 h(Y_b) \\ T_0, X_a, X_b \Vdash_1 h(Y_a) & T_1, S_0, Y_b, h(Y_b) \Vdash_1 X_b, X_a, h(Y_b) \\ T_0, X_a, X_b, h(Y_a) \Vdash_1 X_a, X_b, h(Y_a) & T_2, S_0, S_1 \Vdash_1 K_b \end{cases}$$

Then, we guess an equivalence relation R on $St_E(\mathcal{C}')$

$$R = \{X_a = I; X_b = C; Y_b = K_i; K_b = Y_a \oplus Y_b\}.$$

This equivalence relation represents a unification problem modulo ACUNh. There are finitely many most general unifiers. One of them is:

$$\theta = \{X_a \mapsto I; X_b \mapsto C; Y_b \mapsto K_i; Y_a \mapsto K_b \oplus K_i\}.$$

Lastly, we apply θ on $\mathcal{C}'$. For each constraint $T \Vdash_1 u \in \mathcal{C}'$, we have that $u\theta$ is one-step deducible from $T\theta$. The system of $\mathsf{M_E}$ constraints obtained is empty, hence satisfaisable.

10 Conclusion

Our procedure generalizes previous works on the verification of protocols that involve the *exclusive or* operation [CKRT03,CLS03]. Our solution for solving $\mathsf{M_E}$ constraints is general enough to hold in other similar equational theories since it relies on general algebraic concepts. For instance, unification in Abelian groups with homomorphism has been shown to be finitary by Baader [Baa93] with the help of Gröbner bases. Therefore, our approach should extend to that case, thus generalizing the result [MS05]. Apart from the extension to Abelian groups, a future line of research is to study the complexity of the procedure. Contrary to the results obtained in the empty theory [RT03], there is little hope to get an NP-procedure: the similarity between ACUNh unification and AC unification leads to conjecture that there are exponentially many minimal unifiers and that the size of a minimal unifier may be exponential. Since unifiers are explicitly applied during the resolution of deducibility constraints, this would result in an exponential blow up. Another open question is the case of an encryption algorithm distributing over the *exclusive or*. Although the case of a passive intruder is decidable in this framework [LLT05b], the case of an active intruder seems quite intricate since it amounts to having an infinite number of distinct homomorphisms (one for each term used as a key in the encryption algorithm).

References

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

[BC96] A. Boudet and H. Comon. Diophantine equations, Presburger arithmetic and finite automata. In *Proc. Coll. on Trees in Algebra and Programming (CAAP'96)*, volume 1059 of *LNCS*, pages 30–43. Springer, 1996.

[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.

[CD05] H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In *Proc. 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *LNCS*, pages 294–307, Nara (Japan), 2005. Springer.

[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.

[CDL05] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 2005. To appear.

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'97)*, volume 1233 of *LNCS*, pages 103–118, Konstanz, Germany, 1997.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 261–270, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CL04] H. Comon-Lundh. Intruder theories (ongoing work). In *Proc. 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'04)*, volume 2987 of *LNCS*, pages 1–4, Barcelona (Spain), 2004. Springer.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 271–280, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CLT03] H. Comon-Lundh and R. Treinen. Easy intruder deductions. In *Verification: Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *LNCS*, pages 225–242. Springer, 2003.

[CR05] Y. Chevalier and M. Rusinowitch. Combining intruder theories. In *Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 639–651, Lisbon (Portugal), 2005. Springer.

[Del05] Stéphanie Delaune. Easy intruder deduction problems with homomorphisms. *Information Processing Letters*, 2005. To appear.

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

[DJ04] S. Delaune and F. Jacquemard. A decision procedure for the verification of security protocols with explicit destructors. In *Proc. 11th ACM Conference on Computer and Communications Security (CCS'04)*, pages 278–287, Washington, D.C., USA, 2004. ACM Press.

[DLMS99] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on formal methods in security protocols*, Trento, Italy, 1999.

[DY81] D. Dolev and A.C. Yao. On the security of public key protocols. In *Proc. of the 22nd Symp. on Foundations of Computer Science*, pages 350–357, Nashville (USA, Tennessee, 1981. IEEE Comp. Soc. Press.

[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.

[KKS87] 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):683–690, 1987.

[LLT05a] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In *Proc. 16th International Con-*

ference on Rewriting Techniques and Applications (RTA'05), volume 3467 of *LNCS*, pages 308–322, Nara (Japan), 2005. Springer.

[LLT05b] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for the equational theory of exclusive-or with distributive encryption. Research Report LSV-05-19, Laboratoire Spécification et Vérification, ENS Cachan, France, 2005.

[Low96] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. 2nd International Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*, volume 1055 of *LNCS*, pages 147–166, Berlin (Germany), 1996. Springer.

[McA93] D. A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2):284–303, 1993.

[MS05] J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *Journal of Computer Security*, 13(3):515 – 564, 2005.

[Nar96] P. Narendran. Solving linear equations over polynomial semirings. In *Proc. 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 466–472, New Brunswick, New Jersey, 1996. IEEE Comp. Soc. Press.

[NS78] R. Needham and M. Schroeder. Using encryption for authentification in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.

[RT03] M. Rusinowitch and M. Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theoretical Computer Science*, 1-3(299):451–475, 2003.

[Sim94] G.J. Simmons. Cryptoanalysis and protocol failures. *Communications of the ACM*, 37(11):56–65, 1994.

[TMN89] M. Tatebayashi, N. Matsuzaki, and D. B. Newman. Key distribution protocol for digital mobile communication systems. In *Proc. 9th Annual International Cryptology Conference (CRYPTO'89)*, volume 435 of *LNCS*, pages 324–333, Santa Barbara (California, USA), 1989. Springer.

Appendix A Conservative Solutions

Definition 12. *(decomposed) Let P be a proof of $T \vdash u$. We say that a standard term v is decomposed in P if:*

- *either $v = \langle u_1, u_2 \rangle$ and P contains an instance of* (UL) *or* (UR) *whose premise is labeled with $T \vdash \langle u_1, u_2 \rangle$.*
- *or $v = \{u_1\}_{u_2}$ and P contains an instance of* (D) *whose premises are labeled with $T \vdash \{u_1\}_{u_2}$ and $T \vdash u_2$.*

The following proposition has been proved in [RT03] for the standard Dolev-Yao model. The proof of [RT03] can be transferred in a straightforward way to our intruder model which comprises in addition to the standard rules the rule $(\mathsf{M}_{\mathsf{E}})$. It will be used in Lemma 2 to ensure the existence of a proof of $T \vdash u$ which respects some conditions.

Proposition 2. *Let P be a proof of $T \vdash u$ and P' be a minimal proof of $T \vdash \gamma$. Moreover, assume that P' ends with an instance of* (C)*. Then, there exists a proof of $T \vdash u$ in which γ is never decomposed.*

Proof. The proof can be done by induction on the number of instances of inference rules in P which decompose γ. Base case: If there is no such an instance, then P is the expected proof. Assume there are $n+1$ instances of inference rules in P which decompose γ. We can distinguish two cases depending on the fact that γ is a pair (*i.e.* $\langle \gamma_1, \gamma_2 \rangle$) or a ciphertext (*i.e.* $\{\gamma_1\}_{\gamma_2}$). In the first case, this means that there exists an instance of (UL) (or (UR)) whose premise is $\langle \gamma_1, \gamma_2 \rangle$ and conclusion is γ_1 (or γ_2). From P', we can easily extract a proof P_1 of $T \vdash \gamma_1$ (resp. P_2 of $T \vdash \gamma_2$). Note that P_1 (resp. P_2) does not decompose γ by minimality of P'. Hence, such a proof can be plugged to replace the subproof of $T \vdash \gamma_1$ (resp. $T \vdash \gamma_2$) in P which decompose γ. The second case where $\gamma = \{\gamma_1\}_{\gamma_2}$ is similar. We obtain a proof of $T \vdash u$ which contains less instances of inference rules which decompose γ than P. Hence we can apply the induction hypothesis to conclude. $\square$

Remember that we consider implicitly that terms are kept in normal forms, hence we write $u\sigma$ instead of $u\sigma\downarrow$.

Lemma 2. *Let $\mathcal{C}$ be a well-defined constraint system. If there exists a solution σ to $\mathcal{C}$ then there exists a conservative one.*

Proof. We assume given a linear well-founded ordering $\prec$ on standard terms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that the constant 0 is minimal w.r.t. $\prec$. We shall use below the multi-set extension $\ll$ of $\prec$ to multi-sets of standard ground terms. For sake of notation, given two solutions σ_1 and σ_2 of a constraint system, we write $\sigma_1 \ll \sigma_2$ if and only if $\mathit{Fact}_{\mathsf{E}}(\mathit{img}(\sigma_1)) \ll \mathit{Fact}_{\mathsf{E}}(\mathit{img}(\sigma_2))$. Let σ be a minimal (w.r.t. $\ll$) solution to $\mathcal{C}$.

We reason by contradiction to show that σ is conservative w.r.t. $\mathcal{C}$. Assume that there exists $x \in vars(\mathcal{C})$ and $v_x \in Fact_{\mathsf{E}}(x\sigma)$ such $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$ *i.e.* for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ with $t\sigma =_E v_x$, we have $t \notin St_{\mathsf{E}}(\mathcal{C})$. We will show that under this condition there exists a smaller solution σ' of $\mathcal{C}$. Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ and for each $i \leq k$, let $T_i \Vdash u_i$ be the constraint C_i and $C_i\sigma$ be the constraint obtained from C_i by instantiating (and normalizing) all the terms with σ.

Fact 1 *If $v_x \in St_{\mathsf{E}}(s\sigma)$ for some $s \in T_i$ $(i \leq k)$, then there exists $j < i$ such that $v_x \in St_{\mathsf{E}}(u_j\sigma)$.*

We show this result by contradiction. Assume that $v_x \in St_{\mathsf{E}}(s\sigma)$ for some $s \in T_i$ $(i \leq k)$, and that for all $j < i$, we have $v_x \notin St_{\mathsf{E}}(u_j\sigma)$. Let z be a fresh variable, and ρ be the replacement $\{v_x \mapsto z\}$. Let $\theta := \sigma\rho$. We are going to show that $\mathcal{C}\theta$ is not well-formed, leading to a contradiction with the fact that $\mathcal{C}$ is well-defined. Firstly, since $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$, we have $(\mathcal{C}\sigma)\rho = \mathcal{C}(\sigma\rho)$ $(= \mathcal{C}\theta)$. By hypothesis, $v_x \in St_{\mathsf{E}}(T_i\sigma)$, hence $z \in vars(T_i\theta)$. However, for all $j < i$, we have $z \notin vars(u_j\theta)$ since $v_x \notin St_{\mathsf{E}}(u_j\sigma)$.

This allows us to define: $m = \min\{j \mid v_x \in St_{\mathsf{E}}(u_j\sigma)\}$.

Fact 2 *There exists P' a proof of $T_m\sigma \vdash v_x$ which ends with an instance of* (C).

By hypothesis, there exists a minimal proof P of $T_m\sigma \vdash u_m\sigma$. Firstly, we show that there exists in P a node labeled with $T_m\sigma \vdash v_x$. If P contains a node labeled by $T_m\sigma \vdash v_x$, then it is the expected node. Otherwise, we can find recursively a path in P, from the root up to one leaf, where every node which is labeled by $T_m\sigma \vdash u$ is such that $v_x \in St_{\mathsf{E}}(u)$. Thanks to Fact 1, the existence of such a path leads to a contradiction with the minimality of m. Secondly, by definition of m and thanks to Lemma 1 (locality lemma), the subproof P' of P labeled with $T_m\sigma \vdash v_x$ can not be a decomposition proof (otherwise $v_x \in St_{\mathsf{E}}(T_m\sigma)$). Since v_x is necessarily a standard term, this implies that P' ends with an instance of (C).

Now, we let δ be the replacement $\{v_x \mapsto 0\}$. We will show that $\sigma' := \sigma\delta$ is also a solution of $\mathcal{C}$, which is a contradiction since $\sigma' \ll \sigma$ (v_x is a standard term since it is a factor, hence $0 \prec v_x$). For this purpose, we have to build a proof of each $C_i\sigma'$, $i \leq l$. We distinguish two cases.

1. Case $i < m$: By definition of m, $v_x \notin St_{\mathsf{E}}(C_i\sigma)$. In this case, $(C_i\sigma)\delta = C_i\sigma = C_i\sigma'$, *i.e.* σ' is a solution to C_i.
2. Case $i \geq m$: In the remainder, we are going to show that $\sigma' = \sigma\delta$ is also a solution to $C_i = T_i \Vdash u_i$.

Firstly, we may note that $C_i(\sigma\delta) = (C_i\sigma)\delta$ since by hypothesis $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$. By hypothesis σ is a solution to C_i, this means that we have a proof P of $T_i\sigma \vdash u_i\sigma$. Moreover, Fact 2 ensures the existence of a proof of $T_i\sigma \vdash v_x$ which ends with (C) in P. σ' is a solution of $\mathcal{C}_i$, it is obvious for $i = m$ and we extend the result for $i > m$ by well-definedness of $\mathcal{C}$ (stability by any substitution that $\mathcal{C}$ is well-formed). Now, we can apply Proposition 2 to obtain a proof P_i of $T_i\sigma \vdash u_i\sigma$ in which v_x is never decomposed. We shall build from P_i a proof P_i' of $(T_i\sigma)\delta \vdash (u_i\sigma)\delta$ by replacing every subtree ended by $\dfrac{T_i\sigma \vdash v_1 \ldots T_i\sigma \vdash v_n}{T_i\sigma \vdash v_x}$ (C) with a leaf labeled with $T_i\sigma \vdash v_x$ and then by applying δ to every term of the tree obtained.

Fact 3 *P_i' is a proof of $(T_i\sigma)\delta \vdash (u_i\sigma)\delta$.*

To prove this, we have to show that for every node in P_i' labeled with $T_i\sigma\delta \vdash v_0$ and with n sons labeled respectively by $T_i\sigma\delta \vdash v_1, \ldots T_i\sigma\delta \vdash v_n$, the inference $\dfrac{T_i\sigma\delta \vdash v_1 \ldots T_i\sigma\delta \vdash v_n}{T_i\sigma\delta \vdash v_0}$ is an instance of an inference rule of Figure 1.

We distinguish several cases:

- If the inference is a leaf added by the replacement of an instance of (C) in the construction of P_i' given above, then we have $v_0 = 0$, hence $v_0 \in T_i\sigma\delta$.
- If the inference is not a leaf added by the replacement, then we have a "corresponding" inference in P_i. This means that there exists $\dfrac{T_i\sigma \vdash u_1 \ldots T_i\sigma \vdash u_n}{T_i\sigma \vdash u_0}$ an inference step in P_i such that $v_i = u_i\delta$ for each $0 \leq i \leq n$. Since, by construction of P_i' we know that v_x is never decomposed in P_i and the conclusion of an instance of (C) can not be v_x, we can show by case analysis on the inference rule, that when we apply δ on the inference above, we retrieve another instance of the same inference rule. $\square$

Proposition 3. *Let t be a term and σ a substitution. We have:*

$$St_{\mathsf{E}}(t\sigma) \subseteq St_{\mathsf{E}}(t)\sigma \cup \bigcup_{x \in vars(t)} St_{\mathsf{E}}(x\sigma)$$

Proof. This can be easily proved by structural induction on t. If t is a constant or a variable, it is obvious. Now, assume that t is a standard term, *i.e.* $t = f(t_1, \ldots, t_n)$ with $f \in \mathcal{F} \setminus sig(\mathsf{E})$. We have:

$$\begin{aligned} St_{\mathsf{E}}(t\sigma) &= \{t\sigma\} \cup \textstyle\bigcup_{i=1}^n St_{\mathsf{E}}(t_i\sigma) \\ &\subseteq \{t\sigma\} \cup \textstyle\bigcup_{i=1}^n (St_{\mathsf{E}}(t_i)\sigma \cup \bigcup_{x \in vars(t_i)} St_{\mathsf{E}}(x\sigma)) \text{ by induction hypothesis} \\ &\subseteq St_{\mathsf{E}}(f(t_1, \ldots, t_n)\sigma) \cup \textstyle\bigcup_{x \in vars(\{t_1, \ldots, t_n\}} St_{\mathsf{E}}(x\sigma)) \\ &\subseteq St_{\mathsf{E}}(t)\sigma \cup \textstyle\bigcup_{x \in vars(t)} St_{\mathsf{E}}(x\sigma) \end{aligned}$$

Lastly, if t is not a standard term, then we have $t = C[t_1, \ldots, t_n]$ for some standard terms $t_1, \ldots, t_n$ and an E-context C, and we can do the same reasoning as before. $\square$

Obviously, the proposition above can be extended to any set of terms. Note, however, that the inclusion may be strict.

Example 9. Let $t = x + y$ and $\sigma = \{x \mapsto a; y \mapsto a\}$. We have $St_{\mathsf{E}}(t\sigma) = \{0\}$ whereas $St_{\mathsf{E}}(t)\sigma \cup St_{\mathsf{E}}(\{x\sigma, y\sigma\}) = \{0, a\}$.

Lemma 3. *Given a conservative solution σ of $\mathcal{C} = \{C_1, \ldots, C_k\}$. For each $i \leq k$, there exists a proof P_i of $C_i\sigma$ which involves only terms in $St_{\mathsf{E}}(\mathcal{C})\sigma$.*

Proof. Thanks to Lemma 1 (locality lemma), for each i there exists P_i a minimal proof of $T_i\sigma \vdash u_i\sigma$ which only involves terms in $St_{\mathsf{E}}(\mathcal{C}\sigma)$. Thanks to Proposition 3, we have $St_{\mathsf{E}}(\mathcal{C}\sigma) \subseteq St_{\mathsf{E}}(\mathcal{C})\sigma \cup \bigcup_{x \in vars(\mathcal{C})} St_{\mathsf{E}}(x\sigma)$. Hence, we have:

$$\begin{aligned} St_{\mathsf{E}}(\mathcal{C}\sigma) &\subseteq St_{\mathsf{E}}(\mathcal{C})\sigma \cup \textstyle\bigcup_{x \in vars(\mathcal{C})} St_{\mathsf{E}}(x\sigma) \\ &\subseteq (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma \cup \textstyle\bigcup_{x \in vars(\mathcal{C})} St_{\mathsf{E}}(x\sigma) \\ &\subseteq \bar{S}(\mathcal{C})\sigma && \text{since } \sigma \text{ is conservative w.r.t. } \mathcal{C} \end{aligned}$$

where $\bar{S}(\mathcal{C}) = \{C[t_1, \ldots, t_n] \mid \forall i.\ t_i \in St_E(\mathcal{C}) \setminus vars(\mathcal{C}) \text{ and } C \text{ is an } \mathsf{E}\text{-context}\}$

Let $\dfrac{T_i\sigma \vdash u_1 \ldots T_i\sigma \vdash u_n}{T_i\sigma \vdash u_0}$ be an inference in P_i which is an instance of some rule other than $(\mathsf{M_E})$. We can easily show that for all $j \in \{0, \ldots n\}$, $u_j \in St_{\mathsf{E}}(\mathcal{C})\sigma$. Now, we have to deal with the instance of $(\mathsf{M_E})$. By minimality of P_i, an instance of a rule $(\mathsf{M_E})$ can not be followed by another instance of $(\mathsf{M_E})$ (we could otherwise merge the two instances). Hence, for each premise $T_i\sigma \vdash u$ of an instance of $(\mathsf{M_E})$,

- either $T_i\sigma \vdash u$ is the conclusion of an instance of another inference rule than $(\mathsf{M_E})$,
- or we have $u \in T_i\sigma$.

We have also that, the conclusion $T_i\sigma \vdash u$ of an instance of $(\mathsf{M_E})$ is:

- either the premise of an instance of another inference rule than $(\mathsf{M_E})$,
- or we have $u = u_i\sigma$.

Hence we conclude, that there exists a proof P_i of $T_i\sigma \vdash u_i\sigma$ that involves only terms in $St_{\mathsf{E}}(\mathcal{C})\sigma$. $\square$

Appendix B ACUNh Unification

For a signature Σ consisting of constants, an ACUN symbol $+$ and a homomorphism h, the problem of deciding whether two terms are unifiable has been proved decidable in [Nar96]. Baader gives an algorithm for the unification of several equational theories that involve homomorphism, for instance Abelian groups [Baa93]. These algorithms rely on Gröbner bases and are hard to study from the complexity point of view. Therefore we provide a simpler algorithm for the theory ACUNh which generalizes the procedure of [Nar96]. This algorithm provides a finite complete set of unifiers in the signature Σ and it is very close to the classical AC-unification algorithm, which should allow a precise complexity analysis. Then we show that unification with constant restriction (see [BS96] for details) is finitary. This allows us to use the combination algorithm described in [BS96] to get an algorithm that computes a finite complete set of unifiers in Σ augmented with free symbols.

In this Appendix, Σ denotes $\{+, h, c_1, \ldots, c_m\}$. Our solution for computing a finite complete set of unifiers for a unification problem in Σ follows the same lines as the classical approach for AC-unification.

Appendix B.1 Solution to Linear Diophantine Equations in $\mathbb{Z}/2\mathbb{Z}[h]$

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

$$
\begin{array}{l}
A_{1,1}.X_1 + \ldots + A_{1,n}.X_n = B_1 \\
\ldots \\
A_{m,1}.X_1 + \ldots + A_{m,n}.X_n = B_m
\end{array}
$$

where the $A_{i,j}$'s, B_j's are polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$, and the unknowns are instantiated by polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$. The homogeneous system (H) associated to (E) is obtained by replacing all B_j's by 0.

Fact 4 *A solution to (E) is the sum of a solution to (E) and of a solution to (H).*

Proof. The difference of two solutions to (E) is a solution to (H). □

To a polynomial $P(h) = \sum_{i=0}^{i=n} b_i h^i$ with $b_i \in \{0, 1\}$, we associate the number $nb(P)$ whose representation in base 2 is $b_n \ldots b_0$. This correspondence is one-one. We define a total ordering $<$ on polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$ by $P < P'$ if and only if $nb(P) < nb(P')$. This is a total Noetherian ordering (no infinite decreasing sequence exists). Given a (finite or infinite) set of polynomials S, an element $P \in S$ is minimal if there is no element $P' \in S$ such that $P' < P$. For a tuple of polynomials $\boldsymbol{P} = (P_1, \ldots, P_n)$, we define $Nb(\boldsymbol{P})$ as $(Nb(P_1), \ldots, Nb(P_n))$. The ordering $<$ is extended component-wise, yielding a partial order on tuples of polynomials. From now on, *less than, minimal*,.. are defined with respect to this ordering.

Fact 5 *The number of minimal non-null solutions to a system of equations (E) is finite.*

Proof. We recall the classical Dickson's lemma [Dic13]: every infinite sequence of distinct tuples of natural numbers contains at least two (actually infinitely many) comparable tuples. A non-null solution to (E) is a tuple of polynomials. Let us assume that the number of minimal non-null solutions to (E) is infinite. This yields a 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 $Nb(\boldsymbol{T_1}), Nb(\boldsymbol{T_2}), \ldots$, contradiction. □

Since the sum of two solutions to (H) is a solution to (H), we have that a (non-null) solution to (H) is a linear combination of (non-null) minimal solutions to (H). This justifies the next fact:

Fact 6 *A solution to (E) is the sum of a minimal solution to (E) and of a linear combination of the minimal solutions to (H).*

Appendix B.2 ACUNh Unification is finitary for Σ

Fact 7 *Unification in* ACUNh *is finitary.*

The rest of this section is devoted to the proof of this result. A unification problem is equivalent to an equation[5]:

$$\sum_{i=1}^{i=n} A_i \odot X_i = B$$

where the A_i's are constant polynomials, and $B = \Sigma_{i=1}^{m} B_i \odot c_i$ with B_i a constant polynomial, $i = 1, \ldots, n$. Each variable X_i can be written $\Sigma_{j=1}^{j=m} X_{i,j} \odot c_j$ where $X_{i,j}$ are unknown polynomials of $\mathbb{Z}/2\mathbb{Z}[h]$ for $j = 1, \ldots, m$.

The unification problem is equivalent to the following system of linear Diophantine equations (E) over $\mathbb{Z}/2\mathbb{Z}[h]$:

$$\begin{array}{l} A_1.X_{1,1} + \ldots + A_n.X_{n,1} = B_1 \\ \ldots \\ A_1.X_{1,m} + \ldots + A_n.X_{n,m} = B_m \end{array}$$

Example 10. let $\Sigma = \{h, +, c_1, c_2\}$. Let $s = h(X_1) + X_2 + h(c_1)$ and $t = X_1 + h^2(X_2) + c_2$ be two terms, then the unification of s and t amounts to solving
$h(X_1) + X_2 + h(c_1) = X_1 + h^2(X_2) + c_2$
i.e. $h(X_1) + X_1 + h^2(X_2) + X_2 = h(c_1) + c_2$
By writing $X_1 = X_{1,1} \odot c_1 + X_{1,2} \odot c_2$, $X_2 = X_{2,1} \odot c_1 + X_{2,2} \odot c_2$, this is equivalent to:

$$\begin{cases} h.X_{1,1} + X_{1,1} + h^2.X_{2,1} + X_{2,1} = h \\ h.X_{1,2} + X_{1,2} + h^2.X_{2,2} + X_{2,2} = 1 \end{cases} i.e \begin{cases} (h+1).X_{1,1} + (h^2+1).X_{2,1} = h \\ (h+1).X_{1,2} + (h^2+1).X_{2,2} = 1 \end{cases}$$

Each solution to (E) is the sum of a minimal solution to (E) and of a solution to the associated homogeneous system (H). Given a solution $X_{1,1}, \ldots, X_{n,m}$ of (E), we derive a solution $X_1 = \Sigma_{j=1}^{j=m} X_{1,j} \odot c_j, \ldots, X_n = \Sigma_{j=1}^{j=m} X_{n,j} \odot c_j$ of the unification problem. This means that a solution $X_1, \ldots, X_n$ of the unification problem can be written $X_1 = X_1^{\mu} + X_1^{h}, \ldots, X_n = X_n^{\mu} + X_n^{h}$, where $(X_1^{\mu}, \ldots, X_n^{\mu})$ is derived from a minimal solution to (E) and $(X_1^{h}, \ldots, X_n^{h})$ is derived from a solution to (H). Therefore $(X_1^{h}, \ldots, X_n^{h})$ is a solution to the unification problem $\Sigma_{i=1}^{i=n} A_i \odot X_i = 0$.

[5] for simplicity, we consider only one equation, but the algorithm for a finite set of equations is similar

What remains to do is to compute a complete finite set of unifiers of this unification problem from the set of minimal solutions to the equation $A_1X_1 + \ldots + A_nX_n = 0$ in $\mathbb{Z}/2\mathbb{Z}[h]$. Given a tuple $e = (\epsilon_1, \ldots, \epsilon_n)$ of polynomials of $\mathbb{Z}/2\mathbb{Z}[h]$, and a variable U, we denote by $e \odot U$ the tuple $(\epsilon_1 \odot U, \ldots, \epsilon_n \odot U)$.

Fact 8 *$(X_1, \ldots, X_n)$ is a solution to the unification problem $\Sigma_{i=1}^{i=n} A_i \odot X_i = 0$ if and only if $(X_1, \ldots, X_n) = \Sigma_{i \in I} e_i \odot U_i$ with $I \subseteq \{1, \ldots, p\}$, $U_1, U_2, \ldots$ fresh variables.*

Proof. ($\Rightarrow$) Let $(X_1, \ldots, X_n)$ be a solution to the unification problem $\Sigma_{i=1}^{i=n} A_i \odot X_i = 0$. We write $X_i = \Sigma_{i=1}^{i=m} X_{i,j} \odot c_j$ and we get that $\Sigma_{i=1}^{i=n} A_i.X_{i,j} = 0$ for $j = 1, \ldots, m$. Therefore each element of the sequence $X_{1,j}, \ldots, X_{n,j}$ is a linear combination of the e_i's, i.e. it is an expression $\Sigma_{i \in I} \lambda_{i,j}.e_i$ where $\lambda_{i,j} \in \mathbb{Z}/2\mathbb{Z}[h]$ for $i \in I$. Hence $X_1, \ldots, X_n = \Sigma_{j=1}^{j=m} (\Sigma_{i \in I} (\lambda_{i,j}.e_i) \odot c_j) = \Sigma_{i=1}^{i=p} e_i \odot U_i$ for some U_i.
($\Leftarrow$) Simply decompose U_i as $\Sigma_{i=1}^{i=m} U_{i,j} \odot c_j$ and use the fact that e_i is a solution to the equation. $\square$

Since there is a finite number of minimal solutions to (E), an immediate consequence is that unification is finitary. The last step is to compute the set of minimal solutions.

Appendix B.3 An Effective Unification Algorithm

What we need is a way to compute the minimal solutions to a system of Diophantine equations (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: the first-order theory ACUNh is decidable (for the signature Σ) since it is an *automatic structure* [BG00]. Then we show how to construct an automaton that accepts the set of minimal solutions associated to a unification problem. We denote by $FO(\leq, +)$ the first-order theory of the predicate $\leq$ in the theory ACUNh (for the signature Σ).

Proposition 4. *$FO(\leq, +)$ is decidable.*

Proof. The proof is similar to the automata-based procedure for deciding Presburger arithmetic, and the reader is referred to [BC96] for details.

Case of a Single Constant. Firstly, we consider the case where Σ contains only one constant c. Let s, t be ground terms, then there exists polynomials $P = \Sigma_{i=0}^{i=k} a_i h^i$, $Q = \Sigma_{i=0}^{i=l} b_i h^i$ such that $s = P(h) \odot c$, $t = Q(h) \odot c$. We define the relation $s \leq t$ by $P \leq Q$ *i.e.* $a_k \ldots a_0 \leq b_l \ldots b_0$ (we compare the binary representation of two integers). Since $s = t$ is equivalent to $s \leq t \wedge t \leq s$, we shall consider only the predicate $\leq$.

Like in the classical automata-based decision procedure for Presburger arithmetic, we consider automata accepting the representations of tuples of variables (that we read from the right to the left: $b_0 b_1 \ldots$ with a possible addition of trailing zero's). The automata described in Figure 3 accept:

– all pairs (X, Y) such that $Y \leq X$,
– all triples (X, Y, Z) such that $X = Y + Z$.

Actually these automata are simpler than the corresponding automata for Presburger arithmetic since there is no carry-over to deal with.

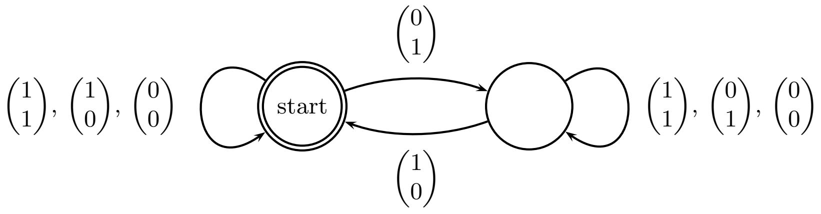

Fig. 3. Automata for $Y \leq X$.

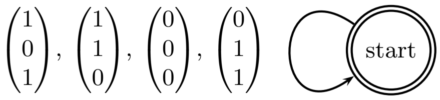

Fig. 4. Automata for $X = Y + Z$.

The automaton accepting the pairs (X, Y) such that $X = h(Y)$ is more complex: it contains two states which remember the previous values of Y. It is described in Figure 5.

The construction of an automaton accepting the terms X such that $X \leq t$ for some ground term t is straightforward. This yields that $FO(\leq, +)$ is decidable if Σ contains only one constant, see [CDG+97] for the description of a similar procedure in the case of Presburger arithmetic.

Case of an Arbitrary Number of Constants. The generalization to several constants is straightforward. Let $s = \Sigma_{i=1}^{i=m} Q_i(h) \odot c_i$ and $t = \Sigma_{i=1}^{i=m} P_i(h) \odot c_i$

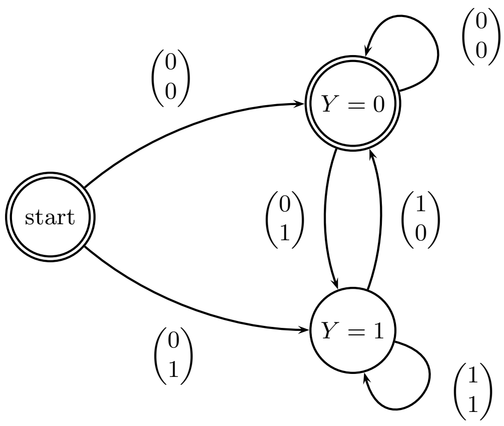

Fig. 5. Automaton for $X = h(Y)$

be two ground terms, Q_i, P_i are polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$. We define $s \leq t$ by $\bigwedge_{i=1}^{i=m} P_i \leq Q_i$. As in the one constant case, we have $s = t$ if and only if $s \leq t \wedge t \leq s$. As previously a polynomial $P(h) = \Sigma_{i=0}^{i=l} b_i h^i$ is identified to the sequence $b_n \ldots b_0$ and a ground term t is identified to the tuple (of representations) $(P_1, \ldots, P_n)$. Therefore the pairs of terms X, Y such that $X \leq Y$ is recognized by the automaton which accepts the 2m-tuples $X_1, \ldots, X_m, Y_1, \ldots, Y_m$ such that $X_i \leq Y_i$. This automaton is simply the product of m automata equals to the first automaton of Figures 3 and 5. The decidability of $FO(\leq, +)$ follows immediately. $\square$

We use this decidability procedure to compute the minimal solutions to a system of Diophantine equations (E).

Fact 9 *The set of minimal solutions to a system of linear Diophantine equations is computable.*

Proof. The unification of two terms leads to a unification problem (U)

$$\sum_{i=1}^{i=n} A_i \odot X_i = B$$

which is equivalent to the system of Diophantine equations (E)

$$\begin{array}{l} A_1.X_{1,1} + \ldots + A_n.X_{n,1} = B_1 \\ \ldots \\ A_1.X_{1,m} + \ldots + A_n.X_{n,m} = B_m \end{array}$$

where $X_i = \Sigma_{j=1}^{j=m} X_{i,j} \odot c_j$ and there is a one-one correspondence between a solution $\boldsymbol{X} = (X_1, \ldots, X_n)$ of the unification problem and a solution to the system of Diophantine equations. We write $\boldsymbol{X} \in Sol(U)$ to denote the fact that $\boldsymbol{X}$ is a solution to (U). This is a formula of the first-order theory $FO(\leq, +)$.

A minimal solution to the system of Diophantine equation corresponds to a solution $\boldsymbol{X} = (X_1, \ldots, X_n)$ of the unification problem such that

$$\forall \boldsymbol{Y}\ \boldsymbol{Y} \leq \boldsymbol{X} \wedge \boldsymbol{Y} \in Sol(U) \implies \boldsymbol{X} = \boldsymbol{Y}$$

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

Appendix B.4 Unification in Σ Augmented with Free Symbols

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 and variables, then we define $V_c = \{X \mid X < c\}$. 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]).

A Technical Result about Unification To prove the next result (Lemma 4), 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 $\mathcal{F} = \Sigma \cup \Sigma'$ 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 are standard, or if C is a context made of symbols of Σ' and the t_i's are not standard.

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 $\#(C[t_1, \ldots, t_n]) = 1 + max\{\#(t_i) \mid i = 1, \ldots, n\}$.

The set $AF(t)$ of alien factors of t is defined by:

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

Lemma 4. *Let P be a unification problem in the theory $\mathsf{E} = \mathsf{ACUNh}$ (including free function symbols) and θ be an mgu_{E} of P. Then for all $x \in dom(\theta)$ and $v \in St_{\mathsf{E}}(x\theta) \setminus vars(x\theta)$ 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 an 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 to 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 $t = C[t_1, \ldots, t_n]$ of P, there exist variables $X_t, X_{t_1}, \ldots, X_{t_n}$ such that $X_t\theta = t\theta = C[X_{t_1}\theta, \ldots X_{t_n}\theta]$.
– For variable $X_{s,t}$, we have $X_{s,t}\theta = s\theta = t\theta$.
– For each term $C[X_1, \ldots, X_l]$ 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 $C[t_1, \ldots, t_l]$ is a alien factor of P.

The solution of the pure unification problems has the form: $X = C'[X_1, \ldots, X_n]$ 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 $CS(P)$. To find the actual set of *mgu*, one must add a last step that detects the unifiers that are subsumed by other elements of $CS(P)$. This step doesn't 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.

Appendix C From Constraints to $\mathsf{M_E}$ Constraints

Lemma 5. *Let $\mathcal{C}$ be a well-defined system of constraints. Let $\mathscr{C}'$ be the set of constraint systems obtained by applying the algorithm of Figure 2 on $\mathcal{C}$.*

1. *$\mathscr{C}'$ is a finite set of well-defined system of one-step constraints.*
2. *Let $\mathcal{C}' \in \mathscr{C}'$. If σ is a solution to $\mathcal{C}'$ then σ is a solution to $\mathcal{C}$.*
3. *If σ is a conservative solution to $\mathcal{C}$ then there exists $\mathcal{C}' \in \mathscr{C}'$ such that σ is a solution to $\mathcal{C}'$.*
4. *For any $\mathcal{C}' \in \mathscr{C}'$, σ is conservative w.r.t. $\mathcal{C}$ if and only if σ is conservative w.r.t. $\mathcal{C}'$.*

Proof. 1. The algorithm described in Figure 2 (see Section 6) is non-deterministic and at each step there are only finitely many possibilities to consider. Hence, $\mathscr{C}'$ is finite. By construction, each constraint system in $\mathscr{C}'$ is a one-step constraint system. Now, let $\mathcal{C}'$ be a one-step constraint system in $\mathscr{C}'$. The monotonicity of $\mathcal{C}'$ is clearly due to the monotonicity of $\mathcal{C}$. To complete the prove of well-definedness of $\mathcal{C}'$ we may observe that each term which appears in an hypothesis set of a constraint is either a term introduced by the algorithm (*i.e.* a term in S) or a term issuing from an hypothesis set of a constraint in $\mathcal{C}$. In the first case, this means that the term appears previously in the target of a constraint by construction. In the second case, we conclude thanks to the well-definedness of $\mathcal{C}$.

2. For each constraint $T_i \Vdash u_i \in \mathcal{C}$, there exists a constraint $T_i \cup S_1 \cup \ldots \cup S_i \Vdash_1 u_i \in \mathcal{C}'$. Since σ is a solution to $\mathcal{C}'$ (by hypothesis), this means that $u_i\sigma$ is one-step deducible from $T_i\sigma \cup S_1\sigma \cup \ldots \cup S_i\sigma$. By construction of $\mathcal{C}'$, we can show that each term in $S_j\sigma$ is deducible by using only terms in $T_j\sigma$. Intuitively, each proof is obtained by stacking "one-step" proofs in a correct order. From this, we easily deduce that $u_i\sigma$ is deducible from $T_1\sigma \cup \ldots \cup T_i\sigma$ which is equal to $T_i\sigma$ thanks to the monotonicity of $\mathcal{C}$.

3. By hypothesis, for each constraint $T_i \Vdash u_i \in \mathcal{C}$, there exists a proof P_i of $T_i\sigma \vdash u_i\sigma$. Since σ is conservative and thanks to Lemma 3 (lifting locality lemma), we can assume that P_i involves only terms in $St_{\mathsf{E}}(\mathcal{C})\sigma$. Let $S'_i = \{s \in St_{\mathsf{E}}(\mathcal{C}) \mid T_i\sigma \vdash s\sigma\}$. In other words, S'_i contains all the subterms of $\mathcal{C}$

whose instance by σ is deducible at step i (*i.e.* by using the terms in T_i). Note that, thanks to the monotonicity of $\mathcal{C}$, we have $S'_i \subseteq S'_{i+1}$ for all $1 \leq i < \ell$. Now, let $S_1 = S'_1$ and $S_i = S'_i \setminus (S'_1 \cup \ldots \cup S'_{i-1})$. S_i contains all the subterms of $\mathcal{C}$ whose instance by σ is deducible at step i and not before. Lastly, for each i, we order the elements in S_i such that: for all $s, s' \in S_i$ such that $T_i\sigma \vdash s\sigma$ is the root of a subproof of a minimal proof of $T_i\sigma \vdash s'\sigma$, then $s \prec_i s'$. Hence, by construction, for each $s \in S_i$, we have that $s\sigma$ is one-step deducible from $S_1\sigma \cup \ldots \cup S_{i-1}\sigma \cup \{s'\sigma \mid s' \prec_i s \text{ and } s' \in S_i\}$. It remains to show that $u_i\sigma$ is one-step deducible from $T_i\sigma \cup S_1\sigma \cup \ldots \cup S_i\sigma$. By definition of the S_j and thanks to the fact that $u_i\sigma$ is deducible at least at step i, we know that $u_i \in S_1 \cup \ldots \cup S_i$. So, we easily deduce that $u_i\sigma \in T_i\sigma \cup S_1\sigma \cup \ldots \cup S_i\sigma$. Hence, the result holds.

4. Let $\mathcal{C}' \in \mathscr{C}'$. We have $St_{\mathsf{E}}(\mathcal{C}') = St_{\mathsf{E}}(\mathcal{C})$. Hence σ is conservative w.r.t. $\mathcal{C}$ if and only if σ is conservative w.r.t. $\mathcal{C}'$. $\square$

Lemma 6. *Given $\mathcal{C}$ a well-defined system of one-step constraints. Let $\mathcal{P} = \{\bigwedge_{(s_1,s_2)\in S'} s_1 = s_2 \mid S' \subseteq St_E(\mathcal{C})^2\}$. Let $R \in \mathcal{P}$ and $\theta \in mgu_{\mathsf{E}}(R)$. Let $\mathcal{C}_\theta = \{T\theta \Vdash_{\mathsf{M_E}} u\theta \mid T \Vdash_1 u \in \mathcal{C}$ and $u\theta$ is not DY-one-step deducible from $T\theta\}$.*

1. *There are only finitely many outputs for a given input $\mathcal{C}$. Each of them is a well-defined system of $\mathsf{M_E}$ constraints.*
2. *If there exists $\mathcal{C}_\theta$ (obtained by the procedure above) which has a solution then $\mathcal{C}$ has a solution.*
3. *If $\mathcal{C}$ has a conservative solution then there exists $\mathcal{C}_\theta$ (obtained by the procedure above) which has a non-collapsing solution.*

Proof. 1. $\mathcal{P}$ is a finite set of equation systems since $St_{\mathsf{E}}(\mathcal{C})$ is finite. Each system of equations represents a unification problem and has a finite complete set of unifiers thanks to Theorem 1. Let θ be such a unifier. Let $\mathcal{C}_\theta$ be a constraint system obtained by using the substitution θ. The system of constraints $\mathcal{C}_\theta$ contains only $\mathsf{M_E}$ constraints. Now, we have to show that $\mathcal{C}_\theta\sigma$ is well-formed for every substitution σ. Let $\mathcal{C}' = \mathcal{C}\theta\sigma$. Thanks to the well-definedness of $\mathcal{C}$, we deduce that $\mathcal{C}'$ is well-formed. It remains to show that the constraints that we need to remove to obtain $\mathcal{C}_\theta$ from $\mathcal{C}'$ do not change anything regarding well-definedness. In other words, we need to show that a removed constraint $T\theta \Vdash_1 u\theta$ does not introduce a variable for the first time, *i.e.* there exists $x \in vars(u\theta)$ and $x \notin vars(T\theta)$. By hypothesis, such a constraint $T\theta \Vdash_1 u\theta$ is such that $u\theta$ is DY-one-step deducible from $T\theta$. Hence $vars(u\theta) \subseteq vars(T\theta)$.

2. Let $\mathcal{C}_\theta$ be the $\mathsf{M_E}$ constraint system obtained from $\mathcal{C}$ by applying the transformation described in the lemma with the substitution θ. Let θ' be a solution to $\mathcal{C}_\theta$. We are going to show that $\theta\theta'$ is a solution to $\mathcal{C}$. Let $T \Vdash_1 u \in \mathcal{C}$. Either $u\theta$ is one-step-deducible from $T\theta$ (without any instantiation) or $T\theta \Vdash_{\mathsf{M_E}} u\theta \in \mathcal{C}'$. In both case, this means that $u\theta\theta'$ is one-step deducible from $T\theta\theta'$. Hence $\theta\theta'$ is a solution of $\mathcal{C}$.

3. Let σ be a conservative solution to $\mathcal{C}$. Let

$$R = \{(s_1, s_2) \mid s_1, s_2 \in St_E(\mathcal{C}) \text{ and } s_1\sigma =_{\mathsf{E}} s_2\sigma\}$$

Let $\theta \in mgu_{\mathsf{E}}(R)$ such that θ is more general than σ. Then, let θ' be the substitution such that $\theta \circ \theta' =_{\mathsf{E}} \sigma$. Let $\mathcal{C}_\theta = \{T\theta \Vdash_{\mathsf{M_E}} u\theta \mid T \Vdash_1 u \in \mathcal{C}$ and $u\theta$ is not DY-one-step deducible from $T\theta\}$. We are going to show that θ' is a solution to $\mathcal{C}_\theta$, *i.e.* $u\theta'$ is $\mathsf{M_E}$-one-step deducible from $T\theta'$ for each $\mathsf{M_E}$ constraint in $\mathcal{C}_\theta$.

Let $T \Vdash_1 u \in \mathcal{C}$ such that $u\sigma$ is DY-one-step deducible from $T\sigma$. We are going to show that $u\theta$ is DY-one-step deducible from $T\theta$. Hence, the one-step constraints which remains in $\mathcal{C}_\theta$ are those that can be solved by using ($\mathsf{M_E}$).

If $u\sigma \in T\sigma$, this means that there exists $t \in T$ such that $t\sigma = u\sigma$. Hence, we have $t\theta = u\theta$ since $t, u \in St_{\mathsf{E}}(\mathcal{C})$. Hence $u\theta \in T\theta$: $u\theta$ is one-step deducible from $T\theta$. Otherwise, $u\sigma$ is one-step deducible from $T\sigma$ by using an inference rule such as (C, UL, UR, D).

In the first case (C), we have $u\sigma = f(v_1, \ldots, v_n)$ for some $v_i \in T\sigma$ and $f \in \mathcal{F} \setminus sig(\mathsf{E})$. Hence, $\forall\ i \leq n\ \exists v'_i \in T$ such that $v_i = v'_i\sigma$. There are two possibilities:

- If u is not a variable, then $u = f(u'_1, \ldots, u'_n)$ and we have $u'_i, v'_i \in St_{\mathsf{E}}(\mathcal{C})$ and $u'_i\sigma = v'_i\sigma$ for each $i \leq n$. Hence, we deduce that $u'_i\theta = v'_i\theta$. Hence $u\theta$ is DY-one-step deducible from $T\theta$.
- If u is a variable, this means (since σ is conservative w.r.t. $\mathcal{C}$) that there exists $t \in St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})$ such that $u\sigma =_{\mathsf{E}} t\sigma$. Hence $t = f(t_1, \ldots, t_n)$ for some $t_i \in St_{\mathsf{E}}(\mathcal{C})$. We can deduce that $t_i = v'_i$. Hence $u\theta$ is DY-one-step deducible from $T\theta$.

The others cases (UR), (UL) and (D) are similar.

We finally have to show that θ' is non-collapsing for $\mathcal{C}_\theta$. Let $u, v \in St_E(\mathcal{C}_\theta) \setminus \mathcal{X}$. Hence, by Proposition 3, $u, v \in St_E(\mathcal{C})\theta \cup \bigcup_{x \in vars(\mathcal{C})} St_E(x\theta)$, by consequence $u, v \in St_E(\mathcal{C})\theta$. By Lemma 4 there are $u_1, v_1 \in St_E(\mathcal{C})$ such that $u = u_1\theta$ and $v = v_1\theta$. Assuming that $u\theta' = v\theta'$, we get:

$$\begin{aligned}
& u\theta' = v\theta' \\
\Rightarrow\ & u_1\theta\theta' = v_1\theta\theta' \\
\Rightarrow\ & u_1\sigma = v_1\sigma \\
\Rightarrow\ & (u_1, v_1) \in R \\
\Rightarrow\ & u_1\theta = v_1\theta \text{ (by construction of } \theta) \\
\Rightarrow\ & u = v
\end{aligned}$$

□

Appendix D Dealing with $\mathsf{M_E}$ Constraints

Proposition 5. *Let $\mathcal{C} = \{t_1, \ldots, t_{n+k-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ be a well-defined $\mathsf{M_E}$ constraint system and σ a non-collapsing solution of $\mathsf{M_E}$. Then for all i, $1 \leq i \leq k$*

1. $Fact_E(u_i\sigma) \subseteq (Fact_E(t_1, \dots, t_{n+i-1}) \setminus \mathcal{X})\sigma$.
2. *For all* x *such that* $i = \min\{j \mid x \in vars(u_j)\}$ *we have that* $Fact_E(x\sigma) \subseteq (Fact_E(t_1, \dots, t_{n+i-1}) \setminus \mathcal{X})\sigma$.

Proof. By induction on the number i of constraints.

If $i = 1$ then $t_1, \dots, t_n$ are ground. We have $Fact_E(u_1\sigma) \subseteq Fact_E(t_1, \dots, t_n)$. Since $Fact_E(t_1, \dots, t_n) \subseteq (Fact_E(t_1, \dots, t_n) \setminus \mathcal{X})\sigma$, we conclude. Moreover, no variable x may occur inside a factor v of u_1, otherwise we would have that $v\sigma = t_g$ for some ground term $t_g \in St_E(t_1, \dots, t_n)$ which contradicts the fact that σ is a non-collapsing substitution. Hence, x must be a factor of u_1, and we conclude as above.

If $i > 1$ then we know that $Fact_E(u_i\sigma) \subseteq (Fact_E(t_1, \dots, t_{n+i-1}))\sigma$. If for some $v \in Fact_E(u_i\sigma)$ we have that $v = x\sigma$ for some $x \in Fact_E(t_j)$ with $1 \leq j < n+i-1$ then we conclude by induction. Now let $x \in vars(u_i\sigma)$, but $x \notin vars(u_j\sigma)$ for $j < i$. Hence, by well-definedness of the constraint system, $x \notin vars(t_1, \dots, t_{n+i-1})$. Hence x cannot occur inside a factor of u_i since this would contradict the non-collapsing of σ. As a consequence, $x \in Fact_E(u_i)$, and we conclude as in the first half of the induction hypothesis. □

Lemma 7. *If a well-defined* $\mathsf{M_E}$*-constraint system* $\mathcal{C}$ *has a non-collapsing solution then it is factor-preserving.*

Proof. We write $T_i = \{t_j \mid 1 \leq j \leq n+i-1\}$. Let σ be a non-collapsing solution of $\mathcal{C}$ and $v \in Fact_E(u_i) \setminus \mathcal{X}$. Thanks to Proposition 5, there exists $v' \in Fact_E(T_i) \setminus \mathcal{X}$ such that $v\sigma = v'\sigma$. Since σ is a non-collapsing substitution, we have $v = v'$. This allows us to conclude. □

Proposition 1. *Let* A *be a matrix* $n \times m$ *over* $\mathbb{Z}/2\mathbb{Z}[h]$ *such that the* n *row vectors are independent* ($n \leq m$) *then:*

$$\exists Q \in \mathbb{Z}/2\mathbb{Z}[h], \forall b \in \mathbb{Z}/2\mathbb{Z}[h]^n, \exists X \in \mathbb{Z}/2\mathbb{Z}[h]^m \ \ A \cdot X = Q \cdot b \tag{1}$$

Moreover, a coefficient Q *is computable as a determinant of a sub matrix of* A.

Proof. Let $\tilde{A}$ be a matrix $m \times m$ obtained from A by adding some rows and such that all the rows of $\tilde{A}$ are independent. Let $Q = det(\tilde{A})$ ($det(A)$ denotes the usual mathematical notion of determinant of the matrix $n \times n$ A over $\mathbb{Z}/2\mathbb{Z}[h]$). We have $det(\tilde{A}) \neq 0$ since all the rows of $\tilde{A}$ are independent and $\tilde{A}^{-1}$ (the inverse of $\tilde{A}$) is equal to $\frac{1}{Q}.A'$ where A' is a matrix with coefficients in $\mathbb{Z}/2\mathbb{Z}[h]$. Let $b \in \mathbb{Z}/2\mathbb{Z}[h]^n$, let $\tilde{b} \in \mathbb{Z}/2\mathbb{Z}[h]^m$ be a vector obtained from b by adding some arbitrary coefficients. $\tilde{X} = \tilde{A}^{-1}.Q.\tilde{b}$ is a particular solution of the over constrained system $\tilde{A} \cdot \tilde{X} = Q.\tilde{b}$. Note that X obtained from $\tilde{X}$ by keeping the n first components is a solution of $A \cdot X = Q.b$. □

Let t be a term with the decomposition $t = t^{X_1} \odot X_1 + \ldots t^{X_n} \odot X_n + t^0$, where $t^{X_i} \in \mathbb{Z}/2\mathbb{Z}[h]$ and t^0 a ground term. The application of a substitution σ to t is calculated as

$$t\sigma = t^{X_1} \odot X_1\sigma + \ldots + t^{X_n} \odot X_n\sigma + t_0$$

This can be seen as the sum of some kind of scalar product and the constant term t^0, that is in matrix notation:

$$t\sigma = (t^{X_1}, \ldots, t^{X_n}) \odot \begin{pmatrix} X_1\sigma \\ \vdots \\ X_n\sigma \end{pmatrix} + t^0$$

This will be used in later in the proofs.

Proposition 6. *Let $t_1, t_2 \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ with $t_1 \neq t_2$. Let $c^X \in \mathbb{Z}/2\mathbb{Z}[h]$ for every $X \in \mathcal{X}$, let $W \notin \mathcal{X}$ be a fresh variable, and let $\theta\colon \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \{W\})$ be a substitution such that $X\theta = X + c^X \odot W$ for every $X \in \mathcal{X}$. Then $t_1\theta \neq t_2\theta$.*

Proof. We proceed by induction on the size of the terms t_1 and t_2. The base case is obvious.

- If t_1 and t_2 are both standard terms then we have

$$\begin{aligned} t_1 &= f_1(t_1^1, \ldots, t_1^n) \\ t_2 &= f_2(t_2^1, \ldots, t_2^m) \end{aligned}$$

  If $f_1 \neq f_2$ then obviously $t_1\theta \neq t_2\theta$.
  If $f_1 = f_2$ then $t_1^i \neq t_2^i$ for some i, hence by induction hypothesis $t_1^i\theta \neq t_2^i\theta$, and $t_1\theta = f_1(t_1^1\theta, \ldots, t_1^n\theta) \neq f_2(t_2^1\theta, \ldots, t_2^n\theta) = t_2\theta$.
- If t_1 is standard and t_2 is not standard then we have

$$\begin{aligned} t_1 &= f_1(t_1^1, \ldots, t_1^n) \\ t_2 &= \Sigma_{s \in Fact_E(t_2)}(p^s \odot s) \end{aligned}$$

  for some polynomials $p^s \in \mathbb{Z}/2\mathbb{Z}[h]$, and $Fact_{\mathsf{E}}(t_2)$ contains at least two elements. By induction hypothesis for all $s_1, s_2 \in Fact_{\mathsf{E}}(t_2)$ with $s_1 \neq s_2$, $s_1\theta \neq s_2\theta$. As a consequence, $t_2\theta$ is not standard, and since $t_1\theta$ is standard we conclude that $t_1\theta \neq t_2\theta$.
- If both t_1 and t_2 are not standard then let $F = Fact_{\mathsf{E}}(t_1) \cup Fact_{\mathsf{E}}(t_2)$. We can decompose t_1 and t_2 as

$$\begin{aligned} t_1 &= \Sigma_{X \in F \cap \mathcal{X}}(p_1^X \odot X) + \Sigma_{s \in F \setminus \mathcal{X}}(p_1^s \odot s) \\ t_2 &= \Sigma_{X \in F \cap \mathcal{X}}(p_2^X \odot X) + \Sigma_{s \in F \setminus \mathcal{X}}(p_2^s \odot s) \end{aligned}$$

  Hence, by definition of θ, we obtain that

$$\begin{aligned} t_1\theta &= \Sigma_{X \in F \cap \mathcal{X}}(p_1^X \odot X) + \Sigma_{X \in F \cap \mathcal{X}}((p_1^X.c^X) \odot W) + \Sigma_{s \in F \setminus \mathcal{X}}(p_1^s \odot s\theta) \\ t_2\theta &= \Sigma_{X \in F \cap \mathcal{X}}(p_2^X \odot X) + \Sigma_{X \in F \cap \mathcal{X}}((p_2^X.c^X) \odot W) + \Sigma_{s \in F \setminus \mathcal{X}}(p_2^s \odot s\theta) \end{aligned}$$

Since $t_1 \neq t_2$ we have two cases:
In the first case, $p_1^X \neq p_2^X$ for some $X \in F \cap \mathcal{X}$, by consequence, since $W \neq X$, we obtain that X occurs in $t_1\theta$ with coefficient p_1^X and in $t_2\theta$ with coefficient p_2^X, hence $t_1\theta \neq t_2\theta$.
In the second case $p_1^s \neq p_2^s$ for some $s \in F \setminus \mathcal{X}$. By induction hypothesis, $s\theta \neq s'\theta$ for every $s \in F \setminus \mathcal{X}$ with $s \neq s'$. As a consequence, $s\theta$ occurs in $t_1\theta$ with coefficient p_1^s and in $t_2\theta$ with coefficient p_2^s, hence $t_1\theta \neq t_2\theta$. $\square$

Remark 1. We can show that such a θ (defined in Proposition 6) do not allow us to cancel distinct standard subterms in a term t. More precisely, if s is a non variable standard subterm of t, then $s\theta$ is also a non variable standard subterm of $t\theta$.

Lemma 8. *Let $\mathcal{C} = \{t_1, \ldots, t_{n+i-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ be a well-defined and factor-preserving $\mathsf{M_E}$ constraint system. Then for every $i \leq k$ and $s \in NSt(t_{n+i-1})$ the set $\{\boldsymbol{s}\} \cup \mathcal{B}_{i-1}$ is dependent.*

Proof. We proceed by induction on i. If $i = 1$ then $t_{n+i-1} = t_n$ is ground, and hence $\boldsymbol{s} = (0, \ldots, 0)$ for every $s \in NSt(t_n)$. We conclude since the set $\{(0, \ldots, 0)\}$ is dependent.

Let $1 < i \leq k$. By contradiction, we suppose that there is a $s \in NSt(t_{n+1-1})$ such that $\{\boldsymbol{s}\} \cup \mathcal{B}_{i-1}$ is independent. We will now construct a substitution θ such that $\mathcal{C}\theta$ is not well-founded. The idea of the construction is to generalize the example 6.

By Proposition 1 there is some $Q \in \mathbb{Z}/2\mathbb{Z}[h], Q \neq 0$, and a vector $(c^{X_1}, \ldots, c^{X_p}) \in (\mathbb{Z}/2\mathbb{Z}[h])^p$ such that

$$\begin{pmatrix} u_1^{X_1} & \ldots & u_1^{X_p} \\ \vdots & & \vdots \\ u_{i-1}^{X_1} & \ldots & u_{i-1}^{X_p} \\ s^{X_1} & \ldots & s^{X_p} \end{pmatrix} \cdot \begin{pmatrix} c^{X_1} \\ \vdots \\ c^{X_p} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ Q \end{pmatrix}$$

where only those row vectors $\boldsymbol{u_j}$ appear in the matrix for which $j \in L$. We define the substitution $\theta\colon \{X_1, \ldots, X_p\} \to \mathcal{T}(F, \{X_1, \ldots, X_p, W\})$ by

$$X\theta = X + c^X \odot W$$

for any $X \in \{X_1, \ldots, X_p\}$.

Claim 1: Let $j < i$. For any subterm v of t_{n+j-1} we have that $W \notin vars(v\theta)$ (*).

If v is a standard term, that is $v = f(v_1, \ldots, v_n)$ with f standard then $v\theta = f(v_1\theta, \ldots, v_n\theta)$ and the claim follows by application of the induction hypothesis (*) to the terms v_i. Otherwise we can decompose:

$$v = \Sigma_{i=1,\ldots,p}(v^{X_i} \odot X_i) + \Sigma_{f \in Fact_E(v) \setminus \mathcal{X}}(v^f \odot f)$$

Hence we obtain by definition of θ that

$$\begin{aligned} v\theta &= \Sigma_{i=1,\ldots,p}(v^{X_i} \odot X_i\theta) + \Sigma_{f\in Fact_E(v)\setminus\mathcal{X}}(v^f \odot f\theta) \\ &= \underbrace{\Sigma_{i=1,\ldots,p}(v^{X_i} \odot X_i)}_{\sigma_1} + \underbrace{\Sigma_{i=1,\ldots,p}((v^{X_i}.c^{X_i}) \odot W)}_{\sigma_2} + \underbrace{\Sigma_{f\in Fact_E(v)\setminus\mathcal{X}}(v^f \odot f\theta)}_{\sigma_3} \end{aligned}$$

Since $W \neq X_1, \ldots, X_p$ we have that $W \notin vars(\sigma_1)$. By induction hypothesis (*) , $W \notin vars(\sigma_3)$. By induction hypothesis (of Lemma 8), the set $\{\boldsymbol{v}\} \cup \mathcal{B}_{j-1}$ is dependent. Hence, there are coefficients $\alpha, \alpha_1, \ldots, \alpha_{j-1} \in \mathbb{Z}/2\mathbb{Z}[h]$ with $\alpha \neq 0$ such that

$$\alpha.\boldsymbol{v} + \alpha_1.\boldsymbol{u_1} + \ldots \alpha_{j-1}.\boldsymbol{u_{j-1}} = (0, \ldots, 0)$$

that is

$$\alpha.\boldsymbol{v} = \alpha_1.\boldsymbol{u_1} + \ldots \alpha_{j-1}.\boldsymbol{u_{j-1}}$$

If we apply scalar multiplication with the vector $(c^{X_1}, \ldots, c^{X_p})$ to both sides of this equation we obtain that

$$\alpha.\Sigma_{i=1}^{p}(v^{X_i}.c^{X_i}) = \alpha_1.\Sigma_{i=1}^{p}(u_1^{X_i}.c^{X_i}) + \ldots + \alpha_{j-1}.\Sigma_{i=1}^{p}(u_{j-1}^{X_i}.c^{X_i})$$

The right hand side of this equation is 0 according to the definition of θ (since in fact each parts of the sum is 0), hence so is the left hand side. Since $\alpha \neq 0$ we conclude that $\Sigma_{i=1}^{p}(v^{X_i}.c^{X_1}) = 0$, that is $\sigma_2 = 0 \odot W = 0$, and $W \notin vars(\sigma_2)$.

As a consequence, $W \notin vars(t_{n+j-1}\theta)$ for any $j < n$.

Claim 2: $W \notin vars(u_j\theta)$ for any $j < i$.

Let us assume to the contrary that $W \in vars(u_j\theta)$. If $j \in L$ then $\Sigma_{i=1}^{p}(u_j^{X_i}.c^{X_i}) = 0$ by construction of θ, and if $j \notin L$ we also have that $\Sigma_{i=1}^{p}(u_j^{X_i}.c^{X_i}) = 0$ since in this case $\boldsymbol{u_j}$ is dependent on $\mathcal{B}_{j-1}$. As a consequence, there is some $f \in Fact_E(u_j\theta) \setminus \mathcal{X}$ such that $W \in vars(f)$. This is only possible if there is a $f' \in Fact(u_j)$ such that $W \in f = f'\theta$. By factor-preservation of the constraint-system $\mathcal{C}$ there is some $j' \leq j < i$ such that $f' \in Fact(t_{n+j'-1})$. By Proposition 6, for every $f'' \in Fact_E(t_{n+j'-1})$ with $f' \neq f''$ we have $f'\theta \neq f''\theta$, hence $W \in vars(t_{n+j'-1}\theta)$, which is a contradiction to claim (1) proved above.

Claim 3: $W \in vars(t_{n+i-1}\theta)$.

We can decompose

$$s = \Sigma_{i=1}^{p}(s^{X_i} \odot X_i) + \Sigma_{f\in Fact(s)\setminus\mathcal{X}}(s^f \odot f)$$

By definition of θ we obtain that

$$s\theta = \Sigma_{i=1}^{p}(s^{X_i} \odot X_i) + \underbrace{\Sigma_{i=1}^{p}((s^{X_i}.c^{X_i}) \odot W)}_{=Q\odot W} + \Sigma_{f\in Fact(s)\setminus\mathcal{X}}(s^f \odot f\sigma)$$

and hence $W \in s\theta$.

If $s = t_{n+i-1}$ then we conclude immediately that $W \in vars(t_{n+i-1}\theta)$. Otherwise, let $f \in Fact_E(t_{n+i-1}) \setminus \mathcal{X}$ such that $s \in NSt(f)$. By Proposition 6, since $f \in Fact_E(t_{n+i-1}) \setminus \mathcal{X}$ hence $f\theta$ can not be canceled, we also have that $W \in vars(f\theta)$, and that $W \in vars(t_{n+i-1}\theta)$.

Hence, $W \in vars(t_{n+i-1}\theta)$ and $W \notin vars(u_j\theta)$ for all $j < i$, which is a contradiction to the origination of $\mathcal{C}\theta$, and hence also a contradiction to the well-definedness of $\mathcal{C}$. $\square$

Lemma 12. *Let $\mathcal{C} = \{t_1, \ldots, t_{n+k-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ be a $\mathsf{M_E}$ constraint system over the signature $\{h, +\} \cup \mathcal{F}_0$, and let $L \subseteq \{1, \ldots, k\}$ be the set of indices computed as in Section 7.2. The system $\mathcal{C}$ is well-defined if and only if for all $1 \leq j \leq k$ the set $\{\boldsymbol{u_i} \mid i \in L \text{ and } i \leq j\} \cup \{\boldsymbol{t_{n+j}}\}$ is dependent.*

Proof. ($\Leftarrow$) Firstly, it is easy to see that monotonicity is clearly satisfied. Secondly, we have to show that for every substitution θ, $\mathcal{C}\theta$ satisfies the origination property. Let θ be a substitution and t be a term which appears in an hypothesis set of $\mathcal{C}$ such that $t\theta$ contains a variable Z. We have necessarily $t = t_{n+j}$ for some j (otherwise t would be a ground term) and we have to show that $Z \in u_i\theta$ for some $i \leq j$. By hypothesis, we know that there exist $\alpha, \alpha_i \in \mathbb{Z}/2\mathbb{Z}[h]$ such that $\sum \alpha_i \boldsymbol{u_i} + \alpha \boldsymbol{t_{n+j}} = 0$ and α, α_i are not all null. Note that $\{\boldsymbol{u_i} \mid i \in L \text{ and } i \leq j\}$ is independent since it is a subset of $\mathcal{B}$. This implies that $\alpha \neq 0$. Hence, we have:

$$\begin{array}{rrcl}
 & \alpha.\boldsymbol{t_{n+j}} &=& -\sum_{i\in L, i\leq j} \alpha_i.\boldsymbol{u_i} \\
\Rightarrow & \alpha.(\sum_{l=1}^{l=p} t_{n+j}^{X_l} + t_{n+j}^0 - t_{n+j}^0) &=& -\sum_{i\in L, i\leq j} \alpha_i.(\sum_{l=1}^{l=p} u_i^{X_l} + u_i^0 - u_i^0) \\
\Rightarrow & \alpha.(t_{n+j} - t_{n+j}^0) &=& -\sum_{i\in L, i\leq j} \alpha_i.(u_i - u_i^0) \\
\Rightarrow & \alpha.(t_{n+j}\theta - t_{n+j}^0) &=& -\sum_{i\in L, i\leq j} \alpha_i.(u_i\theta - u_i^0) \\
\Rightarrow & \alpha.t_{n+j}\theta &=& -\sum_{i\in L, i\leq j} \alpha_i.(u_i\theta - u_i^0) + \alpha t_{n+j}^0
\end{array}$$

Hence, $Z \in vars(t_{n+j}\theta)$ implies that $Z \in vars(u_i\theta)$ for some $i \in L$ and $i \leq j$.

($\Rightarrow$) Assume that there exists $1 \leq j \leq k$ such that $\{\boldsymbol{u_i} \mid i \in L \text{ and } i \leq j\} \cup \{\boldsymbol{t_{n+j}}\}$ is independent. By Proposition 1, there exists $Q \in \mathbb{Z}/2\mathbb{Z}[h]$ such that the following system of equations has a solution over $\mathbb{Z}/2\mathbb{Z}[h]$.

$$\begin{pmatrix} u_1^{X_1} & u_1^{X_2} & \ldots & u_1^{X_p} \\ \vdots & & & \\ u_j^{X_1} & u_j^{X_2} & \ldots & u_j^{X_p} \\ t_{n+j}^{X_1} & t_{n+j}^{X_2} & \ldots & t_{n+j}^{X_p} \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix} = Q. \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Let $(c_1, \ldots, c_p)$ be a solution to this system of equations. Let Z be a fresh variable and θ be the substitution defined by $X_i \mapsto c_i.Z$ for $1 \leq i \leq p$ where Z is a fresh variable. By construction of θ, we have $u_i\theta = u_i^0$ for each $i \in L$ such that $i \leq j$ and $t_{n+j}\theta = Q.Z + t_{n+j}^0$. In other words, we have found a substitution θ such that Z appears for the first time in an hypothesis set of $\mathcal{C}\theta$. This contradicts the well-definedness of $\mathcal{C}$. $\square$

Lemma 9. *Let $\mathcal{C}$ be a well-defined factor-preserving $\mathsf{M_E}$ constraint system and $F = Fact_E(\mathcal{C})\backslash\mathcal{X}$. Let $\mathcal{F}_0$ be a set of new constant symbols of the same cardinality as F and $\rho : F \to \mathcal{F}_0$ a bijection.*

1. *$\mathcal{C}^\rho$ is well-defined.*
2. *$vars(\mathcal{C}^\rho) = vars(\mathcal{C})$.*
3. *If $\mathcal{C}$ has a non-collapsing solution then $\mathcal{C}^\rho$ has a $\mathcal{F}_0 \cup \{h, +\}$-solution.*
4. *If $\mathcal{C}^\rho$ has a $\mathcal{F}_0 \cup \{h, +\}$-solution then $\mathcal{C}$ has a solution.*

Proof. 1. Well-definedness of $\mathcal{C}^\rho$ is a consequence of Lemma 8 and Lemma 12.
2. We have that $vars(\mathcal{C}^\rho) \subseteq vars(\mathcal{C})$ since ρ does not introduce any new variables. Conversely, if $x \in vars(\mathcal{C})$ then let i be the smallest index such that $x \in vars(u_i)$. By the same argument as above, x must have an occurrence in u_i which is not inside a factor, hence $x \in vars(\mathcal{C}^\rho)$.
3. Let σ be a non-collapsing solution to $\mathcal{C}$. By non-collapsing of σ, $v_1\sigma = v_2\sigma$ implies $v_1 = v_2$ and hence $v_1\rho = v_2\rho$ for all $v_1, v_2 \in Fact_E(\mathcal{C}) \setminus \mathcal{X}$. Hence, σ^ρ is a solution to $\mathcal{C}^\rho$.
4. Let σ be a solution to $\mathcal{C}\rho$. Then $\sigma^{(\rho^{-1})}$ is a solution to $\mathcal{C}$. □

Lemma 13. *Let $\mathcal{C} = \{t_1, \ldots, t_{n+k-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ be a well-defined $\mathsf{M_E}$ constraint system over the signature $\{h, +\} \cup F_0$, and let $L \subseteq \{1, \ldots, k\}$ be the set of indices computed as in Section 7.2. Let $N \leq k$ and σ, σ' two substitutions (not necessarily solutions to $\mathcal{C}$). If $u_i\sigma = u_i\sigma'$ for all $1 \leq i \leq N$ with $i \in L$ then we also have that $t_j\sigma = t_j\sigma'$ for all $1 \leq j < n + N$.*

Proof. This is an immediate consequence of Lemma 12 and of the independence of the set $\{\boldsymbol{u}_i \mid i \in L\}$. □

Note that the variables of $\mathcal{Z}$ are totally ordered by the lexicographic order of the indices of variables, that is $z[i,j] \prec z[i',j']$ if and only if $i < i'$, or else $i = i'$ and $j < j'$.

Lemma 14. *Let $\mathcal{S}(\mathcal{C})$ be a system of equations obtained from a well-defined $\mathsf{M_E}$ constraint system $\mathcal{C}$, and let $\mathcal{Z}' \subseteq Z_L$ be downward closed (i.e., if $z_1 \prec z_2 \in \mathcal{Z}'$ then $z_1 \in \mathcal{Z}'$). If $\mathcal{S}(\mathcal{C})$ has a solution then there exists a solution σ to $\mathcal{S}(\mathcal{C})$ such that $0 \leq z\sigma < Q_{max}$ for all $z \in \mathcal{Z}'$.*

Lemma 10 follows from Lemma 14 by choosing $\mathcal{Z}' = \mathcal{Z}_L$.

Proof. We prove this result by induction on the cardinality of $\mathcal{Z}'$.
Base case: $\mathcal{Z}' = \emptyset$ and the result is trivial.
Induction step: let $z = max_\prec(\mathcal{Z}')$. Since $\mathcal{Z}' \subseteq \mathcal{Z}_L$, we have $z = z[N, M]$ for some integers N, M such that $N \in L$ and $1 \leq M < n + N$. By induction hypothesis there exists a solution σ such that $0 \leq z'\sigma < Q_{max}$ for all $z' \in \mathcal{Z}'$ with $z' \neq z$.

In the following, we construct a solution σ' of $\mathcal{S}$ such that $0 \leq z\sigma' < Q_{max}$ for all $\forall z \in \mathcal{Z}'$. The construction of σ' will be in four steps. Finally, we will prove that σ' is indeed a solution to $\mathcal{S}$.

1. Definition of $z'\sigma'$ for $z' \prec z$ We set

$$z'\sigma' = z\sigma \qquad \text{for } z' \in \mathcal{Z}' \setminus \{z\}$$

2. Definition of $z\sigma'$

Note that $z = z[N, M]$. Let $K, r \in \mathbb{Z}/2\mathbb{Z}[h]$ such that $0 \leq r < Q_{max}$ and $z\sigma = r + K \cdot Q_{max}$. We set

$$z[N, M]\sigma' = r$$

3. Definition of $x\sigma'$ for $x \in vars(\mathcal{S})$

Our goal is to find σ' such that:

- for each $i \in L \setminus \{N\}$, $u_i\sigma - u_i\sigma' = 0$
- $u_N\sigma - u_N\sigma' = K \cdot Q_{max} t_M \sigma$

Note that these equations do not involve the context variables from $\mathcal{Z}$, hence we can solve these equations without having yet fixed the value of σ' on all variables from $\mathcal{Z}$. We have to solve the following matrix equation, where the value of a variable X_i' corresponds to $X_i\sigma - X_i\sigma'$:

$$\begin{pmatrix} u_1^{X_1} \; u_1^{X_2} \; \ldots \; u_1^{X_p} \\ \vdots \\ u_N^{X_1} \; u_N^{X_2} \; \ldots \; u_N^{X_p} \\ \vdots \\ u_\ell^{X_1} \; u_\ell^{X_2} \; \ldots \; u_\ell^{X_p} \end{pmatrix} \odot \begin{pmatrix} X_1' \\ \vdots \\ X_p' \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ K \cdot Q_{max} \odot t_M\sigma \\ \vdots \\ 0 \end{pmatrix} \tag{2}$$

This can be achieved by solving the system of equations described below where the unknowns Y_i take values in $\mathbb{Z}/2\mathbb{Z}[h]$:

$$\begin{pmatrix} u_1^{X_1} \; u_1^{X_2} \; \ldots \; u_1^{X_p} \\ \vdots \\ u_N^{X_1} \; u_N^{X_2} \; \ldots \; u_N^{X_p} \\ \vdots \\ u_\ell^{X_1} \; u_\ell^{X_2} \; \ldots \; u_\ell^{X_p} \end{pmatrix} \cdot \begin{pmatrix} Y_1 \\ \vdots \\ Y_p \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ K \cdot Q_{max} \\ \vdots \\ 0 \end{pmatrix} \tag{3}$$

Thanks to Proposition 1 the equation (3) has a solution $(c_1, \ldots, c_p)$. As a consequence, $(c_1 \odot t_M\sigma, \ldots, c_p \odot t_M\sigma)$ is a solution to (2).

This allows us to define σ' on $vars(\mathcal{C})$ by:

$$X_i\sigma' = X_i\sigma - c_i \odot t_M\sigma \qquad i = 1, \ldots, p$$

4. Definition of $z'\sigma'$ for $z \prec z'$

Note that $z \prec z'$ if and only if z' is some $z[i, q]$ with either $i = N$ and $q > M$, or $i > N$. Hence , if $z \prec z[i, q]$ then either $q = M$ and $i > N$, or $q \neq M$.

$$z[i, q]\sigma' = \begin{cases} z[i, q]\sigma + \displaystyle\sum_{j=n+N}^{n+i-1} (\sum_{l=1}^{p} t_j^{X_l} \cdot c_l) \cdot z[i, j]\sigma & \text{if } q = M, i > N \\ z[i, q] & \text{if } q \neq M \end{cases}$$

5. Verification that σ' is a solution to $\mathcal{S}$

First note that

$$t_j\sigma = t_j\sigma' \qquad \text{for } 1 \leq j < n+N \tag{4}$$

This is a consequence of Lemma 13 and of the fact that $u_i\sigma = u_i\sigma'$ for $1 \leq i < N$.

First case $i < N$: This is immediate by (4) and the fact that $u_i\sigma = u_i\sigma'$ for $1 \leq i < N$.

Second case $i = N$: we notice that

$$r = z[N,M]\sigma - K \cdot Q_{max} \tag{5}$$

Hence,

$$\begin{aligned}
&\sum_{j=1}^{n+N-1} z[N,j]\sigma' \odot t_j\sigma' \\
&= \sum_{j=1}^{M-1} z[N,j]\sigma' \odot t_j\sigma' + z[N,M]\sigma' \odot t_M\sigma' + \sum_{j=M+1}^{n+N-1} z[N,j]\sigma' \odot t_j\sigma' \\
&= \sum_{j=1}^{M-1} z[N,j]\sigma \odot t_j\sigma + r \odot t_M\sigma + \sum_{j=M+1}^{n+N-1} z[N,j]\sigma \odot t_j\sigma \\
&\qquad \text{(by (4) and } z[N,j]\sigma = z[N,j]\sigma' \text{ for } j \neq M) \\
&= \sum_{j=1}^{M-1} z[N,j]\sigma \odot t_j\sigma + (z[N,M]\sigma - K \cdot Q_{max}) \odot t_M\sigma + \sum_{j=M+1}^{n+N-1} z[N,j]\sigma \odot t_j\sigma \\
&\qquad \text{(by (5))} \\
&= \sum_{j=1}^{n+N-1} z[N,j]\sigma \odot t_j\sigma - K.Q_{max} \odot t_M\sigma \\
&= u_N\sigma - K.Q_{max} \odot t_M\sigma \\
&\qquad \text{(since } \sigma \text{ is a solution to } \mathcal{S}) \\
&= u_N\sigma' \\
&\qquad \text{(by definition of } \sigma')
\end{aligned}$$

Third case $i > N$: we consider the i-th equation of $\mathcal{S}$ *i.e.*: $\sum_{1 \leq j < n+i} z[i,j] \odot t_j = u_i$ Note that, using $X_i\sigma' = X_i\sigma - c_i \odot t_M\sigma$, we get that:

$$\begin{aligned}
t_j\sigma' &= \sum_{v \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})} (t_j^v \odot v)\sigma' + \sum_{v \in vars(\mathcal{C})} (t_j^v \odot v)\sigma' \\
&= \sum_{v \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})} (t_j^v \odot v) + \sum_{l=1}^{p} (t_j^{X_l} \odot X_l\sigma) - \sum_{l=1}^{p} (t_j^{X_l}.c_l \odot t_M\sigma)
\end{aligned} \tag{6}$$

Hence, we obtain:

$$\sum_{j=1}^{n+i-1} z[i,j]\sigma' \odot t_j\sigma'$$

$$= \sum_{j=1}^{M-1} z[i,j]\sigma' \odot t_j\sigma' + z[i,M]\sigma' \odot t_M\sigma' + \sum_{j=M+1}^{n+N-1} z[i,j]\sigma' \odot t_j\sigma' + \sum_{j=n+N}^{n+i-1} z[i,j]\sigma' \odot t_j\sigma'$$

$$= \sum_{j=1}^{M-1} z[i,j]\sigma \odot t_j\sigma + z[i,M]\sigma' \odot t_M\sigma' + \sum_{j=M+1}^{n+N-1} z[i,j]\sigma \odot t_j\sigma + \sum_{j=n+N}^{n+i-1} z[i,j]\sigma \odot t_j\sigma'$$

(by (4) and $z[N,j]\sigma = z[N,j]\sigma'$ for $j \neq M$)

$$= \sum_{j=1}^{M-1} z[i,j]\sigma \odot t_j\sigma + (z[i,M]\sigma + \sum_{j=n+N}^{n+i-1} (\sum_{l=1}^{p} t_j^{X_l} \cdot c_l) \cdot z[i,j]\sigma) \odot t_M\sigma$$

$$+ \sum_{j=M+1}^{n+N-1} z[i,j]\sigma \odot t_j\sigma$$

$$+ \sum_{j=n+N}^{n+i-1} z[i,j]\sigma \cdot (\sum_{v \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})} (t_j^v \odot v)$$

$$+ \sum_{l=1}^{p} (t_j^{X_l} \odot X_l\sigma) - \sum_{l=1}^{p} (t_j^{X_l} \cdot c_l \odot t_M\sigma))$$

(by the definition of σ' and (6))

$$= \sum_{j=1}^{n+i-1} z[i,j]\sigma \odot t_j\sigma$$

$$= u_i\sigma$$

(since σ is a solution to $\mathcal{S}$)

$$= u_i\sigma'$$

(since $u_i\sigma = u_i\sigma'$ for $i > N$)

□

Lemma 11. *Given $\mathcal{C}$ a well-defined M_{E} constraint system. It is decidable whether $\mathcal{S}(\mathcal{C})$ has a solution.*

Proof. By Lemma 10 we know that if there exists a solution to $\mathcal{S}(\mathcal{C})$ then there also exists a solution in which the values of all variables in $\mathcal{Z}_L \subseteq \mathcal{Z}$ are bounded by Q_{max}. Hence we can simply guess the value of each variable $z[i,j] \in Z_L$ since there is only a finite number of possibilities.

It remains to decide, for any substitution $\rho_1 : \mathcal{Z}_L \to \mathbb{Z}/2\mathbb{Z}[h]$, whether the system $\mathcal{S}(\mathcal{C})\rho_1$ has a solution. To do this we will proceed in two steps:

1. We will construct by induction on the number of constraints a substitution $\theta\colon \mathit{vars}(\mathcal{C}) \to \mathcal{T}(\mathcal{F})$ such that $\theta \cup \rho_1$ is a solution to all constraints of $\mathcal{S}(\mathcal{C})$ with index in L provided that such a substitution exists.
   We will at the same time show that ρ_1 completely determines the value of each term t in $\mathcal{C}$. In other words, we will show that if θ_1 and θ_2 are solutions to $\mathcal{S}(\mathcal{C})$ extending ρ_1 then $t\theta_1 = t\theta_2$ for any term t of $\mathcal{S}(\mathcal{C})$. To do so we will show by induction on the number i of equations in $\mathcal{S}(\mathcal{C})$ that $\mathcal{S}(\mathcal{C})$ may have a solution only if there exists a substitution $\theta : \mathit{vars}(\mathcal{C}) \to \mathcal{T}(\mathcal{F})$ such that the equations corresponding to indexes in L are satisfied.
2. We show how to decide the remaining problem, *i.e* the satisfiability of the equations in $\mathcal{S}(\mathcal{C})\rho_1\theta$ the indexes of which are not in L.

1. Construction of θ: The base case $i = 1$ we have that either $1 \notin L$ and all the terms of the constraint are ground, or otherwise $1 \in L$ and the context variables $z[1,1], \ldots, z[1,n]$ are in $\mathcal{Z}_L$ and have already been substituted by ρ_1. It is easy to deduce the value v_1 of u_1. Now, consider the matching problem $\{u_1 = v_1\}$. Either this matching has a solution allowing us to ensure that there exists a substitution θ such that $u_1\theta = v_1$. Otherwise, the first equation has no solution, and the system is not satisfiable (for this choice of ρ_1). Note that the value v_1 is uniquely determined independently of the choice of θ.

Now, we consider a system of $i + 1$ equations. We distinguish two cases:

Case $i + 1 \in L$**:** We know by induction hypothesis that there exists a substitution θ such that all the equations corresponding to indexes in L and smaller than or equal to i are satisfied. Moreover, since $i + 1 \in L$, the context variables appearing in the left hand side of the $i + 1^{th}$ equation are in $\mathcal{Z}_L$, hence they have already been substituted by ρ_1. Now, it is easy to deduce the value v_{i+1} of u_{i+1}. We consider the matching problem $\{u_j = v_j \mid 1 \leq j \leq i + 1\}$. Either this matching problem has a solution and this allows us to ensure, thanks to the well-definedness of the constraint system, that the value of the term t_{n+i} which is completely determined by the values $v_1, \ldots, v_i$ of $u_1, \ldots, u_i$ is a ground term. Otherwise, the matching problem does not have a solution. This means that we cannot find a solution which extends ρ_1.

Case $i + 1 \notin L$**:** By construction the set $U = \{\boldsymbol{u}_j \mid 1 \leq j \leq i, i \in L\}$ is independent but $U \cup \{\boldsymbol{u}_{i+1}\}$ is not independent. Hence $u_{i+1}\theta$ is ground and we can easily determine this value. The same holds for t_{n+i}.

2. Verify the constraints not in L: So far we have obtained a system of equations in which all the terms are ground. We also know that for each $i \in L$, the i^{th} equation is satisfied. Now, we have to deal with the remaining constraints. To do this, we can view each term as a vector (one component per constant) and we can develop each equation to obtain a system of linear equations over polynomials in $\mathbb{Z}/2\mathbb{Z}[h]$. At the end, we have to check whether a linear system of equations over $\mathbb{Z}/2\mathbb{Z}[h]$ has a solution. This is decidable according to [KKS87]. This allows us to decide whether there exists $\rho_2 : \mathcal{Z} \setminus \mathcal{Z}_L \to \mathbb{Z}/2\mathbb{Z}[h]$ a solution of this system. If this is the case then $\theta \cup \rho_1 \cup \rho_2$ is a solution to $\mathcal{S}(\mathcal{C})$. $\square$