Combining algorithms for deciding knowledge in security protocols*

Mathilde Arnaud[1], Véronique Cortier[2], and Stéphanie Delaune[2]

[1] École Normale Supérieure de Cachan, Computer Science department, France
[2] LORIA, CNRS & INRIA project Cassis, Nancy, France

Abstract. In formal approaches, messages sent over a network are usually modeled by terms together with an equational theory, axiomatizing the properties of the cryptographic functions (encryption, exclusive or, ...). The analysis of cryptographic protocols requires a precise understanding of the attacker knowledge. Two standard notions are usually considered: deducibility and indistinguishability. Those notions are well-studied and several decidability results already exist to deal with a variety of equational theories. However most of the results are dedicated to specific equational theories.
We show that decidability results can be easily combined for any disjoint equational theories: if the deducibility and indistinguishability relations are decidable for two disjoint theories, they are also decidable for their union. As an application, new decidability results can be obtained using this combination theorem.

1 Introduction

Security protocols are paramount in today's secure transactions through public channels. It is therefore essential to obtain as much confidence as possible in their correctness. Formal methods have proved their usefulness for precisely analyzing the security of protocols. Understanding security protocols often requires reasoning about knowledge of the attacker. In formal approaches, two main definitions have been proposed in the literature to express knowledge. They are known as message deducibility and indistinguishability relations.

Most often, the knowledge of the attacker is described in terms of message deducibility [17, 19, 18]. Given some set of messages ϕ representing the knowledge of the attacker and another message M, intuitively the secret, one can ask whether an attacker is able to compute M from ϕ. To obtain such a message he uses his deduction capabilities. For instance, he may encrypt and decrypt using keys that he knows.

This concept of deducibility does not always suffice for expressing the knowledge of an attacker. For example, if we consider a protocol that transmits an encrypted Boolean value (e.g. the value of a vote), we may ask whether an attacker can learn this value by eavesdropping on the protocol. Of course, it is

* This work has been partly supported by the RNTL project POSÉ and the ACI Jeunes Chercheurs JC9005.

completely unrealistic to require that the Boolean true and false are not deducible. We need to express the fact that the two transcripts of the protocol, one running with the Boolean value true and the other one with false are *indistinguishable.* Besides allowing more careful formalization of secrecy properties, indistinguishability can also be used for proving the more involved notion of cryptographic indistinguishability [6, 1, 16]: two sequences of messages are cryptographically indistinguishable if their distributions are indistinguishable to any attacker, that is to any probabilistic polynomial Turing machine.

In both cases, deduction and indistinguishability apply to observations on messages at a particular point in time. They do not take into account the dynamic behavior of the protocol. For this reason the indistinguishability relation is called *static equivalence.* Nevertheless those relations are quite useful to reason about the dynamic behavior of a protocol. For instance, the deducibility relation is often used as a subroutine of many decision procedures [20, 7, 11]. In the applied-pi calculus framework [3], it has been shown that observational equivalence (relation which takes into account the dynamic behavior) coincides with labeled bisimulation which corresponds to checking static equivalences and some standard bisimulation conditions.

Both of these relations rely on an underlying equational theory axiomatizing the properties of the cryptographic functions (encryption, exclusive or, . . .). Many decision procedures have been provided to decide these relations under a variety of equational theories. For instance algorithms for deduction have been provided for exclusive or [11], homomorphic operators [13], Abelian groups with distributive encryption [15] and subterm theories [2]. These theories allow basic equations for functions such as encryption, decryption and digital signature. There are also some results on static equivalence. For instance, a general decidability result to handle the class of subterm convergent equational theories is given in [2]. Also in [2], some abstract conditions on the underlying equational theory are proposed to ensure decidability of deduction and static equivalence. Note that the use of this result requires checking some assumptions, which might be difficult to prove. This result has been applied to several interesting equational theories such as exclusive or, blind signature and other associative and commutative functions.

For all the previous results, decidability is provided for particular fixed theories or for particular classes of theories. In this paper, we provide a general combination result for both deduction and static equivalence: if the deducibility and indistinguishability relations are decidable for two disjoint theories E_1 and E_2 (that is, the equations of E_1 and E_2 do not share any symbol), they are also decidable for their union $\mathsf{E}_1 \cup \mathsf{E}_2$. Our algorithm for combining theories is polynomial (in the DAG-size of the inputs). It ensures in particular that if the deducibility and indistinguishability relations are decidable for two disjoint theories in polynomial time, they are decidable in polynomial time for their union.

The interest of our result is twofold: first, it allows us to obtain new decidability results from any combination of the existing ones: for example, we obtain that static equivalence is decidable for the theory of encryption combined with exclusive or (and also for example with blind signature), which was not known before. Second, our result allows a modular approach. Deciding interesting equational theories that could not be considered before can be done simply by reducing to the decision of simpler and independent theories.

Our combination result relies on combination algorithms for solving unification problem modulo an equational theory [21, 5]. It follows the approach of Chevalier and Rusinowitch [8], who show how to combine decision algorithms for the deducibility problem in the presence of an active attacker. However, they do not consider static equivalence at all, which is needed to express larger classes of security properties. Moreover, even for deduction, they do not state any combination result in the passive case, though this result might be obtained by adapting their proof. Considering static equivalence notoriously involves more difficulties since static equivalence is defined through overall quantification. In particular, proving static equivalence requires a careful understanding of the (infinite) set of equalities satisfied by a sequence of terms.

Outline of the paper. In Section 2 we introduce notation and definitions as well as the two notions of knowledge. Section 3 provides some material for our combination algorithms. Then Sections 4 and 5 are devoted to the study of deduction and static equivalence respectively. In Section 6, we sum up our results and provide new results obtained as a consequence of our main theorems. Due to lack of space, some proofs are omitted. They can be found in [4].

2 Preliminaries

2.1 Basic definitions

A *signature* Σ consists of a finite set of function symbols, such as enc and pair, each with an arity. A function symbol with arity 0 is a constant symbol. Given a signature Σ, an infinite set of names $\mathcal{N}$, and an infinite set of variables $\mathcal{X}$, we denote by $\mathcal{T}(\Sigma)$ (resp. $\mathcal{T}(\Sigma, \mathcal{X})$) the set of *terms* over $\Sigma \cup \mathcal{N}$ (resp. $\Sigma \cup \mathcal{N} \cup \mathcal{X}$). The former is called the set of ground terms over Σ, while the latter is simply called the set of terms over Σ. The concept of names is borrowed from the applied pi calculus [3] and corresponds to the notion of free constant used for instance in [8]. We write $fn(M)$ (resp. $fv(M)$) for the set of names (resp. variables) that occur in the term M. A context C is a term with holes, or (more formally) a linear term. When C is a context with n distinguished variables $x_1, \ldots, x_n$, we may write $C[x_1, \ldots, x_n]$ instead of C in order to show the variables, and when $T_1, \ldots, T_n$ are terms we may also write $C[T_1, \ldots, T_n]$ for the result of replacing each variable x_i with the corresponding term T_i. A *substitution* σ is a mapping from a finite subset of $\mathcal{X}$ called its domain and written $dom(\sigma)$ to $\mathcal{T}(\Sigma, \mathcal{X})$. Substitutions are extended to endomorphisms of $\mathcal{T}(\Sigma, \mathcal{X})$ as usual. We use a postfix notation for their application.

An *equational presentation* $\mathcal{H} = (\Sigma, \mathsf{E})$ is defined by a set E of equations $u = v$ with $u, v \in \mathcal{T}(\Sigma, \mathcal{X})$ and u, v without names. For any equational presentation $\mathcal{H}$, the relation $=_\mathcal{H}$ denotes the equational theory generated by (Σ, E) on $\mathcal{T}(\Sigma, \mathcal{X})$, that is the smallest congruence containing all instances of axioms of E. Abusively, we shall not distinghuish between an equational presentation $\mathcal{H}$ over a signature Σ and a set E of equations presenting it. Hence, we write $M =_{\mathsf{E}} N$ instead of $M =_{\mathcal{H}} N$ when the signature is clear from the context. A theory E is *consistent* if there do not exist two distinct names n_1 and n_2 such that $n_1 =_{\mathsf{E}} n_2$. Note that, in an inconsistent theory, the problem we are interested in, *i.e.* deduction (defined in Section 2.3) and static equivalence (defined in Section 2.4) are trivial.

Example 1. Let Σ_{xor} be the signature made up of the constant symbol 0 and the binary function $\oplus$ and $\mathsf{E}_{\mathsf{xor}}$ be the following set of equations:

$$\begin{array}{cc} x \oplus (y \oplus z) = (x \oplus y) \oplus z & x \oplus 0 = x \\ x \oplus y = y \oplus x & x \oplus x = 0 \end{array}$$

We have that $n_1 \oplus (n_2 \oplus n_1) =_{\mathsf{E}_{\mathsf{xor}}} n_2$.

Definition 1 (syntactic subterm). *The set $St_s(M)$ of* syntactic subterms *of a term M is defined recursively as follows:*

$$St_s(M) = \begin{cases} \{M\} & \text{if } M \text{ is a variable, a name or a constant} \\ \{M\} \cup \bigcup_{i=1}^{\ell} St_s(M_i) & \text{if } M = f(M_1, \ldots, M_\ell) \end{cases}$$

The positions in a term M are defined recursively as usual (*i.e.* sequences of integers with ϵ being the empty sequence). We denote by $M|_p$ the syntactic subterm of M at position p. The term obtained by replacing $M|_p$ by N is denoted $M[N]_p$.

2.2 Assembling terms into frames

At a particular point in time, while engaging in one or more sessions of one or more protocols, an attacker may know a sequence of messages $M_1, \ldots, M_\ell$. This means that he knows each message but he also knows in which order he obtained the messages. So it is not enough for us to say that the attacker knows the set of terms $\{M_1, \ldots, M_\ell\}$. Furthermore, we should distinguish those names that the attacker knows from those that were freshly generated by others and which remain secret from the attacker; both kinds of names may appear in the terms.

In the applied pi calculus [3], such a sequence of messages is organized into a *frame* $\phi = \nu\tilde{n}.\sigma$, where $\tilde{n}$ is a finite set of *restricted* names (intuitively the fresh ones), and σ is a substitution of the form:

$$\{^{M_1}/_{x_1}, \ldots, ^{M_\ell}/_{x_\ell}\} \quad \text{with} \quad dom(\sigma) = \{x_1, \ldots, x_\ell\}.$$

The variables enable us to refer to each M_i and we always assume that the terms M_i are ground. The names $\tilde{n}$ are bound and can be renamed. Moreover names that do not appear in ϕ can be added or removed from $\tilde{n}$. In particular, we can always assume that two frames share the same set of restricted names.

2.3 Deduction

Given a frame ϕ that represents the information available to an attacker, we may ask whether a given ground term M may be deduced from ϕ. Given an equational theory E on Σ, this relation is written $\phi \vdash_{\mathsf{E}} M$ and is axiomatized by the following rules:

$$\frac{}{\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M} \text{ if } \exists x \in dom(\sigma) \text{ s.t. } x\sigma = M \qquad \frac{}{\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} s}\ s \in \mathcal{N} \smallsetminus \tilde{n}$$

$$\frac{\phi \vdash_{\mathsf{E}} M_1 \ \ \ldots \ \ \phi \vdash_{\mathsf{E}} M_\ell}{\phi \vdash_{\mathsf{E}} f(M_1, \ldots, M_\ell)}\ f \in \Sigma \qquad \frac{\phi \vdash_{\mathsf{E}} M}{\phi \vdash_{\mathsf{E}} M'}\ M =_{\mathsf{E}} M'$$

Intuitively, the deducible messages are the messages of ϕ and the names that are not protected in ϕ, closed by equality in E and closed by application of function symbols. Note that ϕ and M might be built on a signature Σ' that possibly contains some additional function symbol not in Σ. When $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$, any occurrence of names from $\tilde{n}$ in M is bound by $\nu\tilde{n}$. So $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$ could be formally written $\nu\tilde{n}.(\sigma \vdash_{\mathsf{E}} M)$. It is easy to prove by induction the following characterization of deduction.

Lemma 1 (characterization of deduction). *Let M be a ground term and $\nu\tilde{n}.\sigma$ be a frame built on Σ'. Then $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$ if and only if there exists a term $\zeta \in \mathcal{T}(\Sigma, \mathcal{X})$ such that $fn(\zeta) \cap \tilde{n} = \emptyset$ and $\zeta\sigma =_{\mathsf{E}} M$. Such a term ζ is a* recipe *of the term M.*

Example 2. Consider the signature $\Sigma_{\mathsf{enc}} = \{\mathsf{dec}, \mathsf{enc}, \mathsf{pair}, \mathsf{proj}_1, \mathsf{proj}_2\}$. The symbols $\mathsf{dec}, \mathsf{enc}$ and pair are functional symbols of arity 2 that represent respectively the decryption, encryption and pairing functions whereas proj_1 and proj_2 are functional symbols of arity 1 that represent the projection function on respectively the first and the second component of a pair. As usual, we may write $\langle x, y\rangle$ instead of $\mathsf{pair}(x, y)$. The equational theory of pairing and symmetric encryption, denoted by $\mathsf{E}_{\mathsf{enc}}$, is defined by the following equations:

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x, \ \ \mathsf{proj}_1(\langle x, y\rangle) = x \ \text{ and } \ \mathsf{proj}_2(\langle x, y\rangle) = y.$$

Let $\phi = \nu k, s_1.\{^{\mathsf{enc}(\langle s_1, s_2\rangle, k)}/_{x_1}, {}^{k}/_{x_2}\}$. We have that $\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} k$, $\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} s_1$ and also that $\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} s_2$. Indeed x_2, $\mathsf{proj}_1(\mathsf{dec}(x_1, x_2))$ and s_2 are recipes of the terms k, s_1 and s_2 respectively.

We say that deduction is decidable for the equational theory (Σ, E) if the following problem is decidable.

Entries A frame ϕ and a term M built on Σ
Question $\phi \vdash_{\mathsf{E}} M$?

2.4 Static equivalence

Deduction does not always suffice for expressing the knowledge of an attacker, as discussed in the introduction. Sometimes, the attacker can deduce exactly the same set of terms from two different frames but he could still be able to tell the difference between these two frames.

Definition 2 (static equivalence). *Let ϕ be a frame built on Σ' and M and N be two terms. We say that M and N are* equal in the frame ϕ *under the theory E, and write $(M =_{\mathsf{E}} N)\phi$, if there exists $\tilde{n}$ such that $\phi = \nu\tilde{n}.\sigma$, $(fn(M) \cup fn(N)) \cap \tilde{n} = \emptyset$ and $M\sigma =_{\mathsf{E}} N\sigma$. We say that two frames $\phi = \nu\tilde{n}.\sigma$ and $\phi' = \nu\tilde{n}.\sigma'$ built on Σ' are* statically equivalent *w.r.t. (Σ, E), and write $\phi \approx_{\mathsf{E}} \phi'$ (or shortly $\phi \approx \phi'$) when*

- *$dom(\phi) = dom(\phi')$, and*
- *for all $M, N \in \mathcal{T}(\Sigma, \mathcal{X})$ we have that $(M =_{\mathsf{E}} N)\phi \Leftrightarrow (M =_{\mathsf{E}} N)\phi'$.*

Example 3. Consider the equational theory $(\Sigma_{\mathsf{enc}}, \mathsf{E}_{\mathsf{enc}})$ provided in Example 2. Let $\phi = \nu k.\sigma$, $\phi' = \nu k.\sigma'$ where $\sigma = \{^{\mathsf{enc}(s_0,k)}/_{x_1}, {}^{k}/_{x_2}\}$, $\sigma' = \{^{\mathsf{enc}(s_1,k)}/_{x_1}, {}^{k}/_{x_2}\}$. Intuitively, s_0 and s_1 could be the two possible (public) values of a vote. We have $\mathsf{dec}(x_1, x_2)\sigma =_{\mathsf{E}_{\mathsf{enc}}} s_0$ whereas $\mathsf{dec}(x_1, x_2)\sigma' \neq_{\mathsf{E}_{\mathsf{enc}}} s_0$. Therefore we have $\phi \not\approx \phi'$. However, note that $\nu k.\{^{\mathsf{enc}(s_0,k)}/_{x_1}\} \approx \nu k.\{^{\mathsf{enc}(s_1,k)}/_{x_1}\}$.

Let (Σ, E) be an equational theory. We define $\mathsf{Eq}_{\mathsf{E}}(\phi)$ to be the set of equations satisfied by the frame $\phi = \nu\tilde{n}.\sigma$ in the equational theory E:

$$\mathsf{Eq}_{\mathsf{E}}(\phi) = \{(M, N) \in \mathcal{T}(\Sigma, \mathcal{X}) \times \mathcal{T}(\Sigma, \mathcal{X}) \mid (M =_{\mathsf{E}} N)\phi\}.$$

We write $\psi \models \mathsf{Eq}_{\mathsf{E}}(\phi)$ if $(M =_{\mathsf{E}} N)\psi$ for any $(M, N) \in \mathsf{Eq}_{\mathsf{E}}(\phi)$.

Checking for static equivalence is clearly equivalent to checking whether the frames satisfy each other equalities.

Lemma 2 (characterization of static equivalence). *Let $\phi_1 = \nu\tilde{n}.\sigma_1$ and $\phi_2 = \nu\tilde{n}.\sigma_2$ be two frames. We have*

$$\phi_1 \approx_{\mathsf{E}} \phi_2 \quad \Leftrightarrow \quad \phi_2 \models \mathsf{Eq}_{\mathsf{E}}(\phi_1) \text{ and } \phi_1 \models \mathsf{Eq}_{\mathsf{E}}(\phi_2).$$

We say that static equivalence is decidable for the equational theory (Σ, E) if the following problem is decidable.

Entries Two frames ϕ_1 and ϕ_2 built on Σ
Question $\phi_1 \approx_{\mathsf{E}} \phi_2$?

3 Material for combination algorithms

We consider two equational theories $\mathcal{H}_1 = (\Sigma_1, \mathsf{E}_1)$ and $\mathcal{H}_2 = (\Sigma_2, \mathsf{E}_2)$ that are disjoint ($\Sigma_1 \cap \Sigma_2 = \emptyset$) and consistent. We denote by Σ the union of the signatures Σ_1 and Σ_2 and by E the union of the equations E_1 and E_2. The *union* of the two equational theories is $\mathcal{H} = (\Sigma, \mathsf{E})$. Note that the equational theories $\mathcal{H}_1$ and $\mathcal{H}_2$ share symbols (namely names) that can be used to represent agent identities, keys or nonces. In other words, two ground terms t_1 and t_2 such that $t_1 \in \mathcal{T}(\Sigma_1)$ and $t_2 \in \mathcal{T}(\Sigma_2)$ may share some symbols. We simply require that $\Sigma_1 \cap \Sigma_2 = \emptyset$, that is intuitively, the two equational theories do not share cryptographic operators.

3.1 Factors, Subterms

We denote by $\mathsf{sign}(\cdot)$ the function that associates to each term M, the signature (Σ_1 or Σ_2) of the function symbol at position ϵ (root position) in M. For $M \in \mathcal{N} \cup \mathcal{X}$, we define $\mathsf{sign}(M) = \bot$, where $\bot$ is a new symbol. The term N is *alien* to M if $\mathsf{sign}(N) \neq \mathsf{sign}(M)$. We now introduce our notion of *subterms*. A similar notion is also used in [8].

Definition 3 (factors, subterms). *Let $M \in \mathcal{T}(\Sigma, \mathcal{X})$. The* factors *of M are the maximal syntactic subterms of M that are alien to M. This set is denoted $Fct(M)$. The set of its* subterms*, denoted $St(M)$, is defined recursively by*

$$St(M) = \{M\} \cup \bigcup_{N \in Fct(M)} St(N)$$

These notations are extended as expected to sets of terms and frames. Sometimes we will use $St(\phi, M)$ instead of $St(\phi) \cup St(M)$.

Let $M \in \mathcal{T}(\Sigma, \mathcal{X})$. The size $|M|$ of a term M is defined $|M| = 0$ if M is a name or a variable and by $1 + \sum_{i=1}^{n} |N_i|$ if $M = C[N_1, \ldots, N_n]$ where C is a context built on Σ_1 (or Σ_2) and $N_1, \ldots N_n$ are the factors of M.

Example 4. Consider the equational theories $\mathsf{E}_{\mathsf{enc}}$ and $\mathsf{E}_{\mathsf{xor}}$. Let M be the term $\mathsf{dec}(\langle n_1 \oplus \langle n_2, n_3\rangle, \mathsf{proj}_1(n_1 \oplus n_2)\rangle, n_3)$. The term $n_1 \oplus \langle n_2, n_3\rangle$ is a syntactic subterm of M alien to M since $\mathsf{sign}(n_1 \oplus \langle n_2, n_3\rangle) = \Sigma_{\mathsf{xor}}$ and $\mathsf{sign}(M) = \Sigma_{\mathsf{enc}}$. We have that $Fct(M) = \{n_1 \oplus \langle n_2, n_3\rangle,\ n_1 \oplus n_2,\ n_3\}$ and $St(M) = Fct(M) \cup \{M,\ n_1,\ n_2,\ \langle n_2, n_3\rangle\}$. Moreover, we have that $|M| = 4$. Indeed, we have that

$$|M| = 1 + |n_1 \oplus \langle n_2, n_3\rangle| + |n_1 \oplus n_2| + |n_3| = 1 + 2 + 1 + 0 = 4$$

This notion of size of terms is quite non-standard and does not correspond to the actual size of a term. It is only used for proving our lemmas by induction. Our complexity results stated later on in the paper relies on the more usual notion of DAG-size.

3.2 Ordered rewriting

Most of the definitions and results in this subsection are borrowed from [9] since we use similar techniques. We consider the notion of *ordered rewriting* defined in [14], which is a useful tool that has been used (*e.g.* [5]) for proving correctness of combination of unification algorithms. Let $\prec$ be a simplification ordering[3] on ground terms assumed to be total and such that the minimum for $\prec$ is a name n_{min} and the constants in Σ are smaller than any non-constant ground term. We define Σ_0 to be the set of the constant symbols of Σ_1 and Σ_2 plus the name n_{min}.

Given a possibly infinite set of equations $\mathcal{O}$ we define the ordered rewriting relation $\rightarrow_{\mathcal{O}}$ by $M \rightarrow_{\mathcal{O}} M'$ if and only if there exists an equation $N_1 = N_2 \in \mathcal{O}$, a position p in M and a substitution τ such that:

$$M = M[N_1\tau]_p, \quad M' = M[N_2\tau]_p \text{ and } N_2\tau \prec N_1\tau.$$

It has been shown (see [14]) that by applying the *unfailing completion procedure* to a set of equations E we can derive a (possibly infinite) set of equations $\mathcal{O}$ such that on ground terms:

1. the relations $=_{\mathcal{O}}$ and $=_{\mathsf{E}}$ are equal on $\mathcal{T}(\Sigma)$,
2. the rewriting system $\rightarrow_{\mathcal{O}}$ is convergent on $\mathcal{T}(\Sigma)$.

Applying unfailing completion to $\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2$, it is easy to notice [5] that the set of generated equations $\mathcal{O}$ is the disjoint union of the two systems $\mathcal{O}_1$ and $\mathcal{O}_2$ obtained by applying unfailing completion procedures to E_1 and to E_2 respectively. The relation $\rightarrow_{\mathcal{O}}$ being convergent on ground terms we can define $M{\downarrow_{\mathsf{E}}}$ (or briefly $M{\downarrow}$) as the unique normal form of the ground term M for $\rightarrow_{\mathcal{O}}$. We denote by $M{\downarrow_{\mathsf{E}_1}}$ (resp. $M{\downarrow_{\mathsf{E}_2}}$) the unique normal form of the ground term M for $\rightarrow_{\mathcal{O}_1}$ (resp. $\rightarrow_{\mathcal{O}_2}$). We can easily prove the following lemmas.

Lemma 3. *Let M be a ground term such that all its factors are in normal form. Then*

- *either $M{\downarrow} \in Fct(M) \cup \{n_{min}\}$,*
- *or $\mathsf{sign}(M) = \mathsf{sign}(M{\downarrow})$ and $Fct(M{\downarrow}) \subseteq Fct(M) \cup \{n_{min}\}$.*

Lemma 4. *Let M be a ground term such that $\mathsf{sign}(M) = \Sigma_i$ $(i = 1, 2)$ and all its factors are in normal form. Then $M{\downarrow} = M{\downarrow_{\mathsf{E}_i}}$.*

3.3 Normalization and replacements

If Π is a set of positions in a term M, we denote by $M[\Pi \leftarrow N]$ the term obtained by replacing all term at a position in Π by N. We denote by $\delta_{N,N'}$ the replacement of the occurrences of N which appears at a subterm position by N'. It is easy to establish the following lemma.

[3] By definition $\prec$ satisfies that for all ground terms M, N_1, N_2, we have $N_1 \prec M[N_1]$ when M is not the empty context and $N_1 \prec N_2$ implies $M[N_1] \prec M[N_2]$.

Lemma 5. *Let M be a ground term such that all its factors are in normal form. Let $N \in Fct(M)$ and N' be a term alien to M. We have that*

$$(M\delta_{N,N'})\downarrow = ((M\downarrow)\delta_{N,N'})\downarrow).$$

Example 5. Consider the equational theories $\mathsf{E}_{\mathsf{enc}}$ and $\mathsf{E}_{\mathsf{xor}}$.
Let $M = \mathsf{dec}(\mathsf{enc}(\langle n_1 \oplus n_2, n_1 \oplus n_2 \oplus n_3\rangle, n_1 \oplus n_2), n_1 \oplus n_2)$, $N = n_1 \oplus n_2$ and $N' = n$. We have that

- $M\delta_{N,N'} = \mathsf{dec}(\mathsf{enc}(\langle n, n_1 \oplus n_2 \oplus n_3\rangle, n), n)$,
- $M\downarrow\delta_{N,N'} = \langle n, n_1 \oplus n_2 \oplus n_3\rangle$.

Hence, we have that $M\delta_{N,N'}\downarrow = M\downarrow\delta_{N,N'}\downarrow = \langle n, n_1 \oplus n_2 \oplus n_3\rangle$.

Let $\rho : F \to \tilde{n}_F$ be a replacement (that is a function) from a finite set of terms F to names $\tilde{n}_F$. Let $F = \{t_1, \dots, t_k\}$ be a set such that whenever t_i is a syntactic subterm of t_j then $i > j$. For any term M, we denote by M^ρ the term obtained by replacing in M (in an order that is consistent with the subterm relation) any subterm N that is equal modulo E to some $N' \in F$ by $\rho(N')$. Formally, $M^\rho = (M\delta_{t_1,\rho(t_1)})\cdots\delta_{t_k,\rho(t_k)}$. This extends in a natural way to sets of terms, substitutions, frames ...

Example 6. Consider the equational theories $\mathsf{E}_{\mathsf{enc}}$ and $\mathsf{E}_{\mathsf{xor}}$ and the term $t = \mathsf{dec}(\langle n_1 \oplus \langle n_1 \oplus n_2, n_3\rangle, \mathsf{proj}_1(n_1 \oplus n_2)\rangle, n_1 \oplus n_2)$. Let ρ_2 be the replacement $\{n_1 \oplus \langle n_1 \oplus n_2, n_3\rangle \to k_1,\ n_1 \oplus n_2 \to k_2\}$. $t^{\rho_2} = \mathsf{dec}(\langle k_1, \mathsf{proj}_1(k_2)\rangle, k_2)$.

4 Combining algorithms for deduction

This section is devoted to the (sketch of) proof of the following theorem.

Theorem 1. *Let (Σ_1, E_1) and (Σ_2, E_2) be two consistent equational theories such that $\Sigma_1 \cap \Sigma_2 = \emptyset$. If deduction is decidable for (Σ_1, E_1) and (Σ_2, E_2) then deduction is decidable for $(\Sigma_1 \cup \Sigma_2, \mathsf{E}_1 \cup \mathsf{E}_2)$.*

Our algorithm consists in reducing the problem to decide whether $\phi \vdash_{\mathsf{E}} M$ ($\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2$) to several deduction problems. Each of them will be solved either in the equational theory E_1 or in the theory E_2. Our procedure first relies on the existence of a *local proof* of $\phi \vdash M$ which involves only terms in $St(\phi, M)$.

Lemma 6 (locality lemma). *Let $\phi = \nu\tilde{n}.\sigma$ be a frame and M be a ground term built on Σ such that terms in ϕ and M are in normal form. If $\phi \vdash_{\mathsf{E}} M$ then there exists a term ζ built on Σ such that $fn(\zeta) \cap \tilde{n} = \emptyset$ and $\zeta\sigma =_{\mathsf{E}} M$, and for all $\zeta' \in St(\zeta)$, we have that*

- *$\zeta'\sigma\downarrow \in St(\phi, \zeta\sigma\downarrow) \cup \{n_{min}\}$, and*
- *$\zeta'\sigma\downarrow \in St(\phi) \cup \{n_{min}\}$ when $\mathsf{sign}(\zeta') \neq \mathsf{sign}(\zeta'\sigma\downarrow)$.*

Example 7. Consider the theory $\mathsf{E} = \mathsf{E}_{\mathsf{enc}} \cup \mathsf{E}_{\mathsf{xor}}$, the term $M = n_2 \oplus n_3$ and the frame $\phi = \nu n_2, n_3.\{^{\mathsf{enc}(\langle n_1 \oplus n_2, n_3\rangle, n_4)}/_{x_1}\}$. We have that $\phi \vdash_{\mathsf{E}} M$. The recipe $\zeta = \mathsf{proj}_1(\mathsf{dec}(x_1, n_4)) \oplus \mathsf{proj}_2(\mathsf{dec}(x_1, n_4)) \oplus n_1$ satisfies the conditions given in Lemma 6.

We also need to decide deducibility in the theory E_1 (resp. E_2) for terms built on $\Sigma_1 \cup \Sigma_2$. Therefore, we show that we can abstract the alien factors by new names.

Lemma 7. *Let ϕ be a frame and M be a ground term built on Σ such that terms in ϕ and M are in normal form. Let $F_2 = \{N \mid N \in St(\phi, M)$ and $\mathsf{sign}(N) = \Sigma_2\}$, $\tilde{n}_{F_2}$ be a set of names, distinct from the names occurring in ϕ and M, of the same cardinality as F_2 and $\rho_2 : F_2 \to \tilde{n}_{F_2}$ be a replacement. We have that*

$$\phi \vdash_{\mathsf{E}_1} M \text{ if and only if } \nu\tilde{n}_{F_2}.(\phi \vdash_{\mathsf{E}_1} M)^{\rho_2}.$$

A similar result holds by inverting the indices 1 and 2.

We show the lemmas above by using Lemmas 3, 4 and 5 stated in Section 3. Then, our algorithm proceeds by saturation of ϕ by the subterms in $St(\phi, M)$ which are deducible either in (Σ_1, E_1) or in (Σ_2, E_2).

Algorithm. Given a frame ϕ and a term M, we saturate ϕ as follows.

- We start with $\phi_0 = \phi \cup \{n_{min}\}$.
- For any term $T \in St(\phi, M)$, if $\nu\tilde{n}_{F_2}.(\phi_k \vdash_{\mathsf{E}_1} T)^{\rho_2}$ or $\nu\tilde{n}_{F_1}.(\phi_k \vdash_{\mathsf{E}_2} T)^{\rho_1}$ where F_1, F_2, ρ_1, ρ_2 are defined like in Lemma 7, we add T in the set of deducible subterms: $\phi_{k+1} = \phi_k \cup \{T\}$.

We start the procedure over until there are no more $T \in St(\phi, M)$ such that $\nu\tilde{n}_{F_2}.(\phi_k \vdash_{\mathsf{E}_1} M)^{\rho_2}$ or $\nu\tilde{n}_{F_1}.(\phi_k \vdash_{\mathsf{E}_1} M)^{\rho_1}$. Let ϕ^* be the saturated set. Using Lemma 6, we can show that ϕ^* contains exactly the set of all deducible subterms of $St(\phi, M)$. We deduce that $\phi \vdash_{\mathsf{E}_1 \cup \mathsf{E}_2} M$ if and only if $M \in \phi^*$.

Example 8. Consider again Example 7, we successively add in the frame the terms $n_1 \oplus n_2$, n_3 and $n_2 \oplus n_3$.

Complexity. Our reduction is polynomial. Our notion of size for terms was introduced for proving our lemmas by induction. It does not correspond to the actual size of a term since our notion of subterms does not take into account intermediate syntactic subterms. In addition, complexity results for deduction and static equivalence are usually given as functions of the DAG-size of the terms. Thus we express the complexity of our procedure as function of the DAG-size. The DAG-size of a term T, denoted $t_{\mathsf{dag}}(T)$, is the number of distinct syntactic subterms. We assume that $\phi \vdash_{\mathsf{E}_i} M$ can be decided in time $f_i(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))$ where $f_i : \mathbb{N} \to \mathbb{R}$, $i \in \{1, 2\}$. Saturating ϕ requires at most $|St(\phi, M)| \leq t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M)$ steps. At each step, we check whether $\nu\tilde{n}_{F_2}.(\phi_k \vdash_{\mathsf{E}_1} T)^{\rho_2}$ or $\nu\tilde{n}_{F_1}.(\phi_k \vdash_{\mathsf{E}_2} T)^{\rho_1}$ for each $T \in St(\phi, M)$. We deduce that ϕ^* can be computed in time $\mathcal{O}((t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))^2[f_1(2(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))) + f_2(2(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M)))])$. In particular, if deciding $\vdash_{\mathsf{E}_i}$ can be done in polynomial time for $i \in \{1, 2\}$ then deciding $\vdash_{\mathsf{E}_1 \cup \mathsf{E}_2}$ is also polynomial.

5 Combination algorithm for static equivalence

This section is devoted to the (sketch of) proof of the following theorem.

Theorem 2. *Let (Σ_1, E_1) and (Σ_2, E_2) be two equational theories such that $\Sigma_1 \cap \Sigma_2 = \emptyset$. If deduction and static equivalence are decidable for (Σ_1, E_1) and (Σ_2, E_2) then static equivalence is decidable for $(\Sigma_1 \cup \Sigma_2, \mathsf{E}_1 \cup \mathsf{E}_2)$.*

We more precisely show that whenever static equivalence is decidable for (Σ_1, E_1) and (Σ_2, E_2) and deduction is decidable for (Σ, E), then static equivalence is decidable for (Σ, E) where $\Sigma = \Sigma_1 \cup \Sigma_2$ and $\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2$. Thanks to our combination result for deduction (Theorem 1), we know it is sufficient for deduction to be decidable for (Σ_1, E_1) and (Σ_2, E_2). Note that the decidability of $\vdash_{\mathsf{E}_i}$ is not necessarily a consequence of the decidability of $\approx_{\mathsf{E}_i}$. The encoding proposed in [2] works only when there exists a free function symbol in Σ_1.

Our decision procedure works as follows. We first add to the frames all their deducible subterms. This is the reason why we require the decidability of $\vdash_{\mathsf{E}}$. Then, we show that to decide whether $\phi_1 \models \mathsf{Eq}_{\mathsf{E}}(\phi_2)$, it is sufficient to check whether $\phi_1 \models \mathsf{Eq}_{\mathsf{E}_1}(\phi_2)$ and $\phi_1 \models \mathsf{Eq}_{\mathsf{E}_2}(\phi_2)$. Lastly, we abstract alien subterms by fresh names in order to reduce the signature.

5.1 Step 1: adding deducible subterms to the frames

Given $\phi_1 = \nu\tilde{n}.\sigma_1$ and $\phi_2 = \nu\tilde{n}.\sigma_2$ such that $dom(\phi_1) = dom(\phi_2)$, we define the frame $\overline{\phi_2}^{\phi_1}$ by extending ϕ_2 with some of its deducible terms: those for which there exists a recipe ζ such that $\zeta\sigma_1{\downarrow}$ is a subterm of ϕ_1.

$$\overline{\phi_2}^{\phi_1} \stackrel{\text{def}}{=} \phi_2 \cup \{^{\zeta_1\sigma_2\downarrow}/_{y_1}, \ldots, ^{\zeta_n\sigma_2\downarrow}/_{y_n}\}.$$

where y_i is a fresh variable and ζ_i is a recipe of t_i in ϕ_1, *i.e.* $\zeta_i\sigma_1{\downarrow} = t_i$ and $fn(\zeta_i) \cap \tilde{n} = \emptyset$ such that:

- $t_i \in St(\phi_1) \cup \{n_{min}\}$, and
- t_i is not in the image of ϕ_1, that is $t_i \neq x\sigma$ for any $x \in dom(\phi_1)$.

Example 9. Let $\mathsf{E} = \mathsf{E}_{\mathsf{enc}}$ and consider the frames $\phi_2 = \nu n_1, n_2.\{^{\mathsf{enc}(n_1,n_2)}/_{x_1}\}$ and $\phi_1 = \nu n_1, n_2.\{^{\langle n_1,n_2\rangle}/_{x_1}\}$. We have that

$$\overline{\phi_2}^{\phi_1} = \nu n_1, n_2.\{^{\mathsf{enc}(n_1,n_2)}/_{x_1}, ^{\mathsf{proj}_1(\mathsf{enc}(n_1,n_2))}/_{y_1}, ^{\mathsf{proj}_2(\mathsf{enc}(n_1,n_2))}/_{y_2}, ^{n_{min}}/_{y_3}\}$$

The three corresponding recipes are $\mathsf{proj}_1(x_1)$, $\mathsf{proj}_2(x_1)$ and n_{min}.

In particular, we have that $\overline{\phi}^{\phi} = \phi \cup \{^{t_1}/_{y_1}, \ldots, ^{t_n}/_{y_n}\}$ where t_i are the deducible subterms of ϕ. When $\overline{\phi}^{\phi} = \phi$, we say that a frame ϕ contains all its deducible subterms.

Example 10. Consider the frame $\phi = \nu n_2, n_3.\{^{\mathsf{enc}(\langle n_1 \oplus n_2, n_3\rangle, n_4)}/_{x_1}\}$ given in Example 7 and let $\mathsf{E} = \mathsf{E}_{\mathsf{enc}} \cup \mathsf{E}_{\mathsf{xor}}$. We have that

$$\overline{\phi}^{\phi} = \nu n_2, n_3.\{^{\mathsf{enc}(\langle n_1 \oplus n_2, n_3\rangle, n_4)}/_{x_1}, {}^{n_1 \oplus n_2}/_{y_1}, {}^{n_2}/_{y_2}, {}^{n_3}/_{y_3}, {}^{n_1}/_{y_4}, {}^{n_4}/_{y_5}, {}^{n_{min}}/_{y_6}\}.$$

The following lemma ensures that extending frames preserves static equivalence.

Lemma 8. *Let ϕ_1 and ϕ_2 be two frames such that $dom(\phi_1) = dom(\phi_2)$. For any frame ψ such that $dom(\psi) = dom(\phi_1)$, we have that*

$$\overline{\phi_2}^{\psi} \models \mathsf{Eq}_{\mathsf{E}}(\overline{\phi_1}^{\psi}) \text{ if and only if } \phi_2 \models \mathsf{Eq}_{\mathsf{E}}(\phi_1).$$

In particular, we deduce that $\phi_1 \approx_{\mathsf{E}} \phi_2$ if and only if $\overline{\phi_1}^{\phi_2} \approx_{\mathsf{E}} \overline{\phi_2}^{\phi_2}$. Since $\overline{\phi_1}^{\phi_2}$ may not contain all its deducible subterms, we need to extend again the frames with the deducible subterms of $\overline{\phi_1}^{\phi_2}$. However, $\overline{(\overline{\phi_2}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})}$ might not contains its deducible subterms anymore. Lemma 9 states that actually, extending a frame preserves the property of containing all its deducible subterms. The proof of this lemma relies on the locality lemma (Lemma 6) stated in Section 4.

Lemma 9. *Let ϕ be a frame such that $\overline{\phi}^{\phi} = \phi$ and ψ be any frame such that $dom(\psi) = dom(\phi)$. Let $\phi' = \overline{\phi}^{\psi}$. We have that ϕ' contains all its deducible subterms, i.e. $\overline{\phi'}^{\phi'} = \phi'$.*

Thanks to Lemma 8, we deduce that deciding whether $\phi_1 \approx_{\mathsf{E}} \phi_2$ is thus equivalent to deciding whether $\overline{(\overline{\phi_1}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})} \approx_{\mathsf{E}} \overline{(\overline{\phi_2}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})}$ where $\overline{(\overline{\phi_1}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})}$ and $\overline{(\overline{\phi_2}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})}$ contain all their deducible subterms.

Computing $\overline{\phi}^{\psi}$. To compute $\overline{\phi}^{\psi}$, we need to compute the set of deducible subterms of ψ. Moreover, for each deducible subterm T of ψ, we also need to compute a recipe ζ_T such that $(\zeta_T =_{\mathsf{E}} T)\psi$. Such a recipe can usually be deduced from the decision algorithm applied to $\psi \vdash_{\mathsf{E}} T$ [2]. However, if it is not the case, once we know that $\psi \vdash_{\mathsf{E}} T$ (using the decision algorithm), we can enumerate all the recipes until we find ζ such that $(\zeta_T =_{\mathsf{E}} T)\psi$.

5.2 Step 2: Checking for equalities in $\mathsf{Eq}_{\mathsf{E}_i}$

Checking for $\phi \approx_{\mathsf{E}} \psi$ is equivalent to checking for $\phi \models \mathsf{Eq}_{\mathsf{E}}(\psi)$ and $\psi \models \mathsf{Eq}_{\mathsf{E}}(\phi)$. We show that checking for $\psi \models \mathsf{Eq}_{\mathsf{E}}(\phi)$ can actually be done using only equalities in E_1 and E_2.

Proposition 1. *Let ϕ and ψ be two frames such that $\overline{\phi}^{\phi} = \phi$. We have that $\psi \models \mathsf{Eq}_{\mathsf{E}}(\phi)$ if and only if $\psi \models \mathsf{Eq}_{\mathsf{E}_1}(\phi)$ and $\psi \models \mathsf{Eq}_{\mathsf{E}_2}(\phi)$.*

It is straightforward that $\psi \models \mathsf{Eq}_{\mathsf{E}}(\phi)$ implies $\psi \models \mathsf{Eq}_{\mathsf{E}_1}(\phi)$ and $\psi \models \mathsf{Eq}_{\mathsf{E}_2}(\phi)$.

The converse is more difficult. We first introduce some ordering on pairs of terms. We have $(M, N) < (M', N')$ if

$$(\max(|M|, |N|), |M| + |N|) <_{lex} (\max(|M'|, |N'|), |M'| + |N'|)$$

where $<_{lex}$ is the lexicographic order. Now, assuming that $\psi \models \mathsf{Eq}_{\mathsf{E}_1}(\phi)$ and $\psi \models \mathsf{Eq}_{\mathsf{E}_2}(\phi)$, we show by induction on the order on (M, N) that $(M, N) \in \mathsf{Eq}_{\mathsf{E}}(\phi)$ implies $(M, N) \in \mathsf{Eq}_{\mathsf{E}}(\psi)$. The key lemma for the induction step is as follows.

Lemma 10. *Let ϕ and ψ be two frames such that $\overline{\phi}^{\phi} = \phi$, $\psi \models \mathsf{Eq}_{\mathsf{E}_1}(\phi)$ and $\psi \models \mathsf{Eq}_{\mathsf{E}_2}(\phi)$. Let $(M, N) \in \mathsf{Eq}_{\mathsf{E}}(\phi)$ and assume that for all terms M', N'*

$$(M', N') < (M, N) \quad \textit{implies} \quad (M' =_{\mathsf{E}} N')\phi \Rightarrow (M' =_{\mathsf{E}} N')\psi.$$

Let $\phi = \nu\tilde{n}.\sigma$ such that $(fn(M) \cup fn(N)) \cap \tilde{n} = \emptyset$. If there exists $\zeta \in St(M)$ such that $\mathsf{sign}(\zeta\sigma) \neq \mathsf{sign}(\zeta\sigma\downarrow)$, then there exists M_1 such that $|M_1| < |M|$, $(M =_{\mathsf{E}} M_1)\phi$ and $(M =_{\mathsf{E}} M_1)\psi$.

5.3 Step 3: Abstraction of alien subterms

Since ψ and ϕ are built on Σ (and not on Σ_i), we cannot check whether $\psi \approx_{\mathsf{E}_i} \phi$ using the decision algorithm for $\approx_{\mathsf{E}_i}$. We show however that we can simply abstract the alien subterms by fresh names.

Lemma 11. *Let ϕ and ψ be two frames built on Σ and in normal form. Let $F_2 = \{N \in St(\phi \cup \psi) \mid \mathsf{sign}(N) = \Sigma_2\}$, $\tilde{n}_{F_2}$ be a set of names, distinct from the names occurring in ϕ and ψ, of same cardinality as F_2 and $\rho_2 : F_2 \to \tilde{n}_F$ a replacement. We have that*

$$\phi \models \mathsf{Eq}_{\mathsf{E}_1}(\psi) \quad \textit{if and only if} \quad \nu\tilde{n}_{F_2}.\phi^{\rho_2} \models \mathsf{Eq}_{\mathsf{E}_1}(\nu\tilde{n}_{F_2}.\psi^{\rho_2})$$

A similar result holds when inverting the indices 1 and 2.

5.4 Combination algorithm for static equivalence

To sum up, checking for $\phi_1 \approx_{\mathsf{E}} \phi_2$ is performed in two steps:

1. Computing $\phi'_1 = \overline{(\overline{\phi_1}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})}$ and $\phi'_2 = \overline{(\overline{\phi_2}^{\phi_2})}^{(\overline{\phi_1}^{\phi_2})}$.
2. checking for $\nu\tilde{n}_{F_2}.(\phi'_1)^{\rho_2} \approx_{\mathsf{E}_1} \nu\tilde{n}_{F_2}.(\phi'_2)^{\rho_2}$ and $\nu\tilde{n}_{F_1}.(\phi'_1)^{\rho_1} \approx_{\mathsf{E}_2} \nu\tilde{n}_{F_1}.(\phi'_2)^{\rho_1}$.

Complexity. The complexity of the procedure mostly depends on the complexity of computing ϕ'_1 and ϕ'_2 and on their size. In particular, it depends on the time for computing recipes and on their size. Assume that

- $\phi \vdash_{\mathsf{E}} M$ can be decided in $f_3(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))$,
- a recipe ζ such that $(\zeta =_{\mathsf{E}} M)\phi$ can be computed in $f_4(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))$ and that we control the size of the recipe $t_{\mathsf{dag}}(\zeta) \leq f_5(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))$

– $\phi \approx_{\mathsf{E}_i} \psi$ can be decided in $f_i(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))$ for $i \in \{1, 2\}$.

Then it is easy to check that $\phi \approx_{\mathsf{E}} \psi$ can be decided in time polynomial in the $f_i(t_{\mathsf{dag}}(\phi) + t_{\mathsf{dag}}(M))$ with $i \in \{1, \ldots, 5\}$. In particular, if the f_i are polynomial, $\approx_{\mathsf{E}}$ is decidable in polynomial time.

6 Application to new decidability results

Deduction and static equivalence are decidable in polynomial time (in the DAG-size of the inputs) for any convergent subterm theory [2]. A convergent subterm theory is an equational theory induced by a finite set of equations of the form $u = v$ where v is a subterm of u or v is a constant and such that the associate rewriting system is convergent. For example, $\mathsf{E}_{\mathsf{enc}}$ is a convergent subterm theory. From [12], we also know that deduction and static equivalence are decidable in polynomial time for the equational theory $\mathsf{E}_{\mathsf{xor}}$ of the exclusive or and also for the theory E_{AG} of Abelian group. Applying Theorems 1 and 2, we get the following new decidability result.

Proposition 2. *Let* E *be a convergent subterm theory. Deduction and static equivalence are decidable in polynomial time for* $\mathsf{E} \cup \mathsf{E}_{\mathsf{xor}}$ *and* $\mathsf{E} \cup \mathsf{E}_{\mathsf{AG}}$.

Since deduction and static equivalence are also decidable for the theories of blind signature, homomorphic encryption, exclusive or, and other associative-commutative functions [2], we get that deduction and static equivalence are decidable for any combination of these theories.

As further work, we consider extending our combination result for non disjoint theories. This would allow us to consider some fragments of the modular exponentiation theory such as the Diffie-Hellman one, *i.e.* the axioms $\mathsf{exp}(x, 1) = x$ and $\mathsf{exp}(\mathsf{exp}(x, y), z) = \mathsf{exp}(x, y \times z)$ where $\times$ is an Abelian group operator; or to take into account the equation $\mathsf{exp}(x, y) \cdot \mathsf{exp}(x, z) = \mathsf{exp}(x, y + z)$. We might use for example a notion of hierarchy between theories like in [10].

References

1. M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In *Proceedings of the 9th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'06)*, pages 398–412, 2006.
2. M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 387(1-2):2–32, November 2006.
3. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proceedings of the 28th ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM, 2001.
4. M. Arnaud, V. Cortier, and S. Delaune. Combining algorithms for deciding knowledge in security protocols. Research Report 6118, INRIA, Feb. 2007. 28 pages.
5. F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *Journal of Symbolic Computation*, 21(2):211–243, 1996.

6. M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 652–663, Lisboa (Portugal), 2005. Springer-Verlag.
7. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proceedings of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.
8. Y. Chevalier and M. Rusinowitch. Combining intruder theories. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 639–651, Lisbon (Portugal), 2005. Springer.
9. Y. Chevalier and M. Rusinowitch. Combining intruder theories. Technical Report 5495, INRIA, 2005. http://www.inria.fr/rrrt/rr-5495.html.
10. Y. Chevalier and M. Rusinowitch. Hierarchical combination of intruder theories. In *Proceedings of the 17th International Conference on Rewriting Techniques and Applications, (RTA'06)*, volume 4098 of *LNCS*, pages 108–122, Seattle (WA), 2006. Springer.
11. H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proceedings of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.
12. V. Cortier and S. Delaune. Deciding knowledge in security protocols for monoidal equational theories. In *Proc. of the Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'07)*, Wrocław, Poland, 2007. To appear.
13. S. Delaune. Easy intruder deduction problems with homomorphisms. *Information Processing Letters*, 97(6):213–218, Mar. 2006.
14. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science*, volume B, chapter 6. Elsevier, 1990.
15. P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for the equational theory of Abelian groups with distributive encryption. *Information and Computation*, 2007. To appear.
16. Y. Lakhnech, L. Mazaré, and B. Warinschi. Soundness of symbolic equivalence for modular exponentiation. In *Proceedings of the Second Workshop on Formal and Computational Cryptography (FCC'06)*, pages 19–23, Venice, Italy, July 2006.
17. G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proceedings of the 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-Verlag.
18. J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS'01)*. ACM Press, 2001.
19. L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1-2):85–128, 1998.
20. 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.
21. M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *Journal of Symbolic Computation*, 8(1/2):51–99, 1989.