YAPA: A generic tool for computing intruder knowledge

Mathieu Baudet
MLstate, France
Véronique Cortier
LORIA - CNRS, France
Stéphanie Delaune
LSV, ENS Cachan & CNRS & INRIA Saclay Ile-de-France, France

Reasoning about the knowledge of an attacker is a necessary step in many formal analyses of security protocols. In the framework of the applied pi calculus, as in similar languages based on equational logics, knowledge is typically expressed by two relations: deducibility and static equivalence. Several decision procedures have been proposed for these relations under a variety of equational theories. However, each theory has its particular algorithm, and none has been implemented so far.

We provide a generic procedure for deducibility and static equivalence that takes as input any convergent rewrite system. We show that our algorithm covers most of the existing decision procedures for convergent theories. We also provide an efficient implementation, and compare it briefly with the tools ProVerif and KiSs.

Categories and Subject Descriptors: F.3.1 [**Logics and Meanings of Programs**]: Verifying and Reasoning about Programs

General Terms: Security

Additional Key Words and Phrases: formal proofs, security protocols, verification, deduction, static equivalence

1. INTRODUCTION

Understanding security protocols often requires reasoning about the information accessible to an on-line attacker. Accordingly, many formal approaches to security rely on a notion of *deducibility* [Lowe 1996; Millen and Shmatikov 2001] that models whether a piece of data, typically a secret, is retrievable from a finite set of messages. Deducibility, however, does not always suffice to reflect the knowledge of

Author's address: S. Delaune, Laboratoire Spécification & Vérification - 61, avenue du président Wilson - 94 230 Cachan.
The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 258865, project ProSecure, and the ANR project JCJC VIP n^o 11 JS02 006 01. A large part of it was done while the first author was working at the ANSSI.
Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 20YY ACM 1529-3785/20YY/0700-0001 $5.00

an attacker. Consider for instance a protocol sending an encrypted Boolean value, say, a vote in an electronic voting protocol. Rather than deducibility, the key idea to express confidentiality of the plaintext is that an attacker should not be able to *distinguish* between the sequences of messages corresponding to each possible value. (Such security considerations typically motivate the use of randomized encryption.)

In the framework of the applied pi-calculus [Abadi and Fournet 2001], as in similar languages based on equational logics [Blanchet et al. 2008], indistinguishability corresponds to a relation called *static equivalence*: roughly, two sequences of messages are *statically equivalent* when they satisfy the same algebraic relations from the attacker's point of view. Static equivalence plays an important role in the study of guessing attacks (e.g. [Corin et al. 2004; Baudet 2005; Abadi et al. 2006]), as well as for anonymity properties and electronic voting protocols (e.g. [Delaune et al. 2009]). Static equivalence is also used for specifying privacy in the context of RFID protocols [Arapinis et al. 2009]. In several cases, this notion has also been shown to imply the more complex and precise notion of cryptographic indistinguishability [Baudet et al. 2005; Abadi et al. 2006], related to probabilistic polynomial-time Turing machines. Two sequences of messages are *cryptographically indistinguishable* when their corresponding bit-string implementations are indistinguishable to any probabilistic polynomial-time Turing machine.

We emphasize that both deducibility and static equivalence apply to observations on finite sets of messages, and do not take into account the dynamic behavior of protocols. (This justifies the expression *static equivalence*.) Nevertheless, deducibility is used as a subroutine by many general decision procedures [Comon-Lundh and Shmatikov 2003a; Chevalier et al. 2003b]. Besides, it has been shown that observational equivalence in the applied pi-calculus coincides with labeled bisimulation [Abadi and Fournet 2001], that is, corresponds to checking an infinite family of static equivalences and some standard bisimulation conditions.

Deducibility and static equivalence rely on an underlying equational theory for axiomatizing the properties of cryptographic functions. Many decision procedures [Abadi and Cortier 2006; Cortier and Delaune 2007] have been proposed to compute these relations under a variety of equational theories, including symmetric and asymmetric encryptions, signatures, exclusive OR, and homomorphic operators. However, except for the class of subterm convergent theories [Abadi and Cortier 2006], which covers the standard flavors of encryption and signature, each of these decision results introduces a new procedure, devoted to a particular theory. Even in the case of the general decidability criterion given in [Abadi and Cortier 2006], we note that the algorithm underlying the proof has to be adapted for each theory, depending on how the criterion is fulfilled.

Perhaps as a consequence of this fact, none of these decision procedures has been implemented so far. When we began this work, the only tool able to verify static equivalence was ProVerif [Blanchet 2001; Blanchet et al. 2008]. This general tool can handle various equational theories and analyze security protocols under active adversaries. However termination of the verifier is not guaranteed in general, and protocols are subject to (safe) approximations. Since then, a new tool, called KiSs, has been developed [Ciobâcă et al. 2009]. The procedure implemented in KiSs has many concepts in common with a preliminary version of this work [Baudet et al.

2009] but targets a different class of equational theories.

The present work aims to fill this gap between theory and implementation and propose an efficient tool for deciding deducibility and static equivalence in a uniform way. It is initially inspired from a procedure for solving more general constraint systems related to active adversaries and equivalence of finite processes, presented in [Baudet 2005], with corrected version in [Baudet 2007] (in French). However, due to the complexity of the constraint systems, this decision procedure was only studied for subterm convergent theories, and remains too complex to enable an efficient implementation.

Our Contributions. In this paper, we provide and study a generic procedure for checking deducibility and static equivalence, taking as input any convergent theory (that is, any equational theory described by a finite convergent rewrite system). We prove the algorithm sound and complete, up to explicit failure cases. Note that (unfailing) termination cannot be guaranteed in general since the problem of checking deducibility and static equivalence is undecidable, even for convergent theories [Abadi and Cortier 2006]. To address this issue and turn our algorithm into a decision procedure for a given convergent theory, we provide two criteria. First, we define a syntactic criterion on the rewrite rules that ensures that the algorithm never fails. This criterion is enjoyed in particular by any convergent subterm theory, as well as the theories of blind signature and homomorphic encryption. Termination often follows from a simple analysis of the rules of the algorithm: as a proof of concept, we obtain a new decidability result for static equivalence for the prefix theory, representing encryption in CBC mode. Moreover, we obtain that our algorithm can decide deducibility and static equivalence for all the convergent theories shown to be decidable in [Abadi and Cortier 2006]. Second, we provide a termination criterion based on deducibility: provided that failure cannot occur, termination on a given input is equivalent to the existence of some natural finite representation of deducible terms.

Our second contribution is an efficient implementation of this generic procedure, called YAPA. After describing the main features of the implementation, we report several experiments suggesting that our tool computes static equivalence faster and for more convergent theories than the general tool ProVerif [Blanchet 2001; Blanchet et al. 2008]. We also outline the main differences between YAPA and the recent tool KiSs.

Related work. Static equivalence is a key notion for equivalence-based properties such as anonymity and other privacy-like properties. To our knowledge, the tools YAPA, KiSs, and ProVerif are the only ones for checking static equivalence for various equational theories. There are however several tools for deciding various trace-based properties (e.g. secrecy and authentication), against active adversaries. Several of these tools are surveyed in [Comon and Shmatikov 2002; Cremers et al. 2009]. We briefly describe here tools that can handle algebraic properties. CL-Atse [Turuani 2006], integrated to the AVISPA platform [Armando et al. 2005], can take into account XOR or several properties of modular exponentiation. The Open-Source Fixed-Point Model Checker (formally named OFMC) [Mödersheim and Viganò 2009] allows one to analyze security protocols with respect to the alge-

braic theory of the employed cryptographic operators, provided they can be specified as part of the input, *i.e.* using some kind of deduction system. TA4SP [Boichut et al. 2006] enables to analyse protocols for an unbounded number of sessions, over-approximating algebraic properties. The Maude-NRL Protocol Analyzer can reason on various algebraic properties of the functions used in a protocol such as the associativity of the pair, one-time pads and Diffie-Hellman [Escobar et al. 2008]. [Goubault-Larrecq et al. 2004] describes an automatic tool for analysing Diffie-Hellman-like protocols. While ProVerif can handle a wide range of algebraic properties, it does not perform very well on non convergent theories, in particular in the case of the XOR operator. Küsters and Truderung have designed and implemented an algorithm for checking trace properties on protocols using the XOR operator [Küsters and Truderung 2010]. This tool takes as input Horn clauses with XOR (modeling the protocols and the security property) and translates them into Horn clauses (without XOR) that can be better handled by ProVerif.

More generally, many decision procedures have been proposed for analysing security protocols in the presence of algebraic properties, without necessarily an implementation. For example, secrecy has been shown decidable for the exclusive or, for a bounded number of sessions [Comon-Lundh and Shmatikov 2003b; Chevalier et al. 2003b] and for an unbounded number of sessions, for particular classes of protocols [Comon-Lundh and Cortier 2003; Verma 2003; Cortier et al. 2007; Seidl and Verma 2009]. In the context of a bounded number of sessions, similar results have been obtained for modular exponentiation and also for the prefix theory [Chevalier et al. 2003b; 2003a; Shmatikov 2004]. A more exhaustive description of the results can be found in [Cortier et al. 2006a].

Outline. We introduce our setting in Section 2, in particular the notion of term algebra and equational theory, that are used to model cryptographic primitives. Deducibility and static equivalence are defined in Section 3. We describe our procedure in Section 4 and prove its correctness and completeness in Section 5. We provide criteria for preventing failure in Section 6 and for ensuring termination in Section 7. The implementation of our procedure is discussed in Section 8. Some concluding remarks and perspectives can be found in Section 9. A number of technical proofs have been postponed to the appendix to ease the presentation.

2. PRELIMINARIES

2.1 Term algebra

We start by introducing the necessary notions to describe cryptographic messages in a symbolic way. For modeling cryptographic primitives, we assume given a set of *function symbols* $\mathcal{F}$ together with an arity function $\mathrm{ar} : \mathcal{F} \to \mathbb{N}$. Symbols in $\mathcal{F}$ of arity 0 are called *constants*. We will denote these constants by a, b, ..., k, ... We consider a set of *variables* $\mathcal{X}$ (denoted by $x, x_1, x_2, \ldots, y, z, \ldots$) and a set of additional constants $\mathcal{W}$ called *parameters* and denoted by $\mathsf{w}, \mathsf{w}_1, \mathsf{w}_2, \ldots$ The (usual, first-order) term algebra generated by $\mathcal{F}$ over $\mathcal{W}$ and $\mathcal{X}$ is written $\mathcal{F}[\mathcal{W} \cup \mathcal{X}]$ with elements denoted by $T, U, T_1 \ldots$ More generally, we write $\mathcal{F}'[A]$ for the least set of terms containing a set A and stable by application of symbols in $\mathcal{F}' \subseteq \mathcal{F}$.

We write $\mathrm{var}(T)$ (resp. $\mathrm{par}(T)$) for the set of variables (resp. parameters) that occur in a term T. These notations are extended to tuples and sets of terms in the

usual way. The set of positions of a term T is written $\mathrm{pos}(T) \subseteq \mathbb{N}^*$, and its set of subterms $\mathrm{st}(T)$. The subterm of T at position $p \in \mathrm{pos}(T)$ is written $T|_p$. The term obtained by replacing $T|_p$ with a term U in T is denoted $T[U]_p$.

A *(finite, partial) substitution* σ is a mapping from a finite subset of variables, called its *domain* and written $\mathrm{dom}(\sigma)$, to terms. The *image* of a substitution is its image as a mapping $\mathrm{im}(\sigma) = \{\sigma(x) \mid x \in \mathrm{dom}(\sigma)\}$. Substitutions are extended to endomorphisms of $\mathcal{F}[\mathcal{X} \cup \mathcal{W}]$ as usual. We use a postfix notation for their application. A term T (resp. a substitution σ) is *ground* if $\mathrm{var}(T) = \emptyset$ (resp. $\mathrm{var}(\mathrm{im}(\sigma)) = \emptyset$).

For our cryptographic purposes, it is useful to distinguish a subset $\mathcal{F}_{\mathsf{pub}}$ of $\mathcal{F}$, made of *public function symbols*, that is, intuitively, the symbols made available to the attacker. The other symbols in $\mathcal{F} \setminus \mathcal{F}_{\mathsf{pub}}$ are private symbols that can not be used by the attacker. This is useful for instance to model a private key constructor in a public key encryption scheme. A *recipe* (or *second-order term*) M, N, $M_1 \ldots$ is a term in $\mathcal{F}_{\mathsf{pub}}[\mathcal{W} \cup \mathcal{X}]$, that is, a term containing no *private* (non-public) function symbols. A *plain term* (or *first-order term*) t, r, s, $t_1 \ldots$ is a term in $\mathcal{F}[\mathcal{X}]$, that is, containing no parameters. A *(public, ground, non-necessarily linear) n-ary context* C is a recipe in $\mathcal{F}_{\mathsf{pub}}[\mathsf{w}_1, \ldots, \mathsf{w}_n]$, where we assume a fixed countable subset of parameters $\{\mathsf{w}_1, \ldots, \mathsf{w}_n, \ldots\} \subseteq \mathcal{W}$. If C is a n-ary context, $C[T_1, \ldots, T_n]$ denotes the term obtained by replacing each occurrence of w_i with T_i in C.

2.2 Rewriting

A *rewrite system* $\mathcal{R}$ is a finite set of *rewrite rules* $l \to r$ where $l, r \in \mathcal{F}[\mathcal{X}]$ and such that $\mathrm{var}(r) \subseteq \mathrm{var}(l)$. A term S *rewrites* to T by $\mathcal{R}$, denoted $S \to_{\mathcal{R}} T$, if there exist $l \to r$ in $\mathcal{R}$, $p \in \mathrm{pos}(S)$ and a substitution σ such that $S|_p = l\sigma$ and $T = S[r\sigma]_p$. We write $\to_{\mathcal{R}}^+$ for the transitive closure of $\to_{\mathcal{R}}$, $\to_{\mathcal{R}}^*$ for its reflexive and transitive closure, and $=_{\mathcal{R}}$ for its reflexive, symmetric and transitive closure.

A rewrite system $\mathcal{R}$ is *convergent* if it is:

—*terminating*, i.e. there is no infinite chain $T_1 \to_{\mathcal{R}} T_2 \to_{\mathcal{R}} \ldots$; and

—*confluent*, i.e. for every terms S, T such that $S =_{\mathcal{R}} T$, there exists U such that $S \to_{\mathcal{R}}^* U$ and $T \to_{\mathcal{R}}^* U$.

A term T is *$\mathcal{R}$-reduced* if there is no term S such that $T \to_{\mathcal{R}} S$. If $T \to_{\mathcal{R}}^* S$ and S is $\mathcal{R}$-reduced then S is *a $\mathcal{R}$-reduced form of T*. When this reduced form is unique (in particular if $\mathcal{R}$ is convergent), we write $S = T{\downarrow_{\mathcal{R}}}$ (or simply $T{\downarrow}$ when $\mathcal{R}$ is clear from the context).

2.3 Equational theories

We equip the signature $\mathcal{F}$ with an equational theory represented by a set of equations $\mathcal{E}$ of the form $s = t$ with $s, t \in \mathcal{F}[\mathcal{X}]$. The equational theory E generated by $\mathcal{E}$ is the least set of equations containing $\mathcal{E}$ that is stable under the axioms of congruence (reflexivity, symmetry, transitivity, application of function symbols) and under application of substitutions. We write $=_{\mathsf{E}}$ for the corresponding relation on terms. Equational theories have proved very useful for modeling algebraic properties of cryptographic primitives (see e.g. [Cortier et al. 2006b] for a survey).

We are particularly interested in theories E that can be represented by a convergent rewrite system $\mathcal{R}$, i.e. theories for which there exists a convergent rewrite system $\mathcal{R}$ such that the two relations $=_{\mathcal{R}}$ and $=_{\mathsf{E}}$ coincide. The rewrite system $\mathcal{R}$ —and by extension the equational theory E— is *weakly subterm convergent* if, in addition, we have that for every rule $l \to r \in \mathcal{R}$, r is either a subterm of l or a ground $\mathcal{R}$-reduced term. This class encompasses the class of subterm convergent theories used in [Abadi and Cortier 2006] (for every rule $l \to r \in \mathcal{R}$, r is a subterm of l or a constant), the class of dwindling theories used in [Anantharaman et al. 2007], and the class of public-collapsing theories introduced in [Delaune and Jacquemard 2004].

Example 2.1. Consider the signature $\mathcal{F}_{\mathsf{enc}} = \{\mathsf{dec}, \mathsf{enc}, \langle \_, \_ \rangle, \mathsf{proj}_1, \mathsf{proj}_2\}$. The symbols $\mathsf{dec}, \mathsf{enc}$ and $\langle \_, \_ \rangle$ 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 the first and the second component of a pair, respectively. The equational theory of pairing and symmetric (deterministic) encryption, denoted by $\mathsf{E}_{\mathsf{enc}}$, is generated by the equations

$$\mathcal{E}_{\mathsf{enc}} = \{\mathsf{dec}(\mathsf{enc}(x, y), y) = x, \;\; \mathsf{proj}_1(\langle x, y \rangle) = x, \;\; \mathsf{proj}_2(\langle x, y \rangle) = y\}.$$

Motivated by the modeling of the ECB mode of encryption, we may also consider an encryption symbol that is homomorphic with respect to pairing:

$$\mathcal{E}_{\mathsf{hom}} = \mathcal{E}_{\mathsf{enc}} \cup \left\{ \begin{array}{rcl} \mathsf{enc}(\langle x, y \rangle, z) & = & \langle \mathsf{enc}(x, z), \mathsf{enc}(y, z) \rangle \\ \mathsf{dec}(\langle x, y \rangle, z) & = & \langle \mathsf{dec}(x, z), \mathsf{dec}(y, z) \rangle \end{array} \right\}.$$

If we orient the equations from left to right, we obtain two rewrite systems $\mathcal{R}_{\mathsf{enc}}$ and $\mathcal{R}_{\mathsf{hom}}$ that represent respectively the theories $\mathsf{E}_{\mathsf{enc}}$ and $\mathsf{E}_{\mathsf{hom}}$, i.e. $=_{\mathcal{R}_{\mathsf{enc}}}$ and $=_{\mathsf{E}_{\mathsf{enc}}}$ (resp. $=_{\mathcal{R}_{\mathsf{hom}}}$ and $=_{\mathsf{E}_{\mathsf{hom}}}$) coincide. Both rewrite systems are convergent, only $\mathcal{R}_{\mathsf{enc}}$ is (weakly) subterm convergent. Other examples of subterm convergent theories can be found in [Abadi and Cortier 2006].

From now on, we assume given an equational theory E represented by a convergent rewrite system $\mathcal{R}$. A symbol f is *free* if f does not occur in $\mathcal{R}$. In order to model (an unbounded number of) random values possibly generated by the attacker, we assume that $\mathcal{F}_{\mathsf{pub}}$ contains infinitely many free public constants. We will use free private constants to model secrets, for instance the secret keys used to encrypt a message. Private (resp. public) free constants are closely related to bound (resp. free) *names* in the framework of the applied pi calculus [Abadi and Fournet 2001]. Our formalism also allows one to consider non-constant private symbols.

3. DEDUCIBILITY AND STATIC EQUIVALENCE

In order to describe the cryptographic messages observed or inferred by an attacker, we introduce the following notions of deduction facts and frames.

A *deduction fact* is a pair, written $M \rhd t$, made of a recipe $M \in \mathcal{F}_{\mathsf{pub}}[\mathcal{W} \cup \mathcal{X}]$ and a plain term $t \in \mathcal{F}[\mathcal{X}]$. Such a deduction fact is *ground* if $\mathrm{var}(M, t) = \emptyset$. A *frame*, denoted by letters φ, Φ, $\Phi_0 \ldots$, is a finite set of ground deduction facts. The

image of a frame is defined by $\mathrm{im}(\Phi) = \{t \mid M \rhd t \in \Phi\}$. A frame Φ is *one-to-one* if $M_1 \rhd t$, $M_2 \rhd t \in \Phi$ implies $M_1 = M_2$.

A frame φ is *initial* if it is of the form $\varphi = \{w_1 \rhd t_1, \ldots, w_\ell \rhd t_\ell\}$ for some distinct parameters $w_1, \ldots, w_\ell \in \mathcal{W}$. The parameters w_i can be seen as labels that refer to the messages observed by an attacker. Initial frames are closely related to the notion of frames in the applied pi-calculus [Abadi and Fournet 2001]. The only difference is that, in initial frames, values initially unknown to an attacker are modeled by private constants while they are modeled by *restricted names* in the applied pi-calculus. Name generation and binding are important features of the (general) applied calculus but are unessential when considering finite processes, and in particular frames. Given such an initial frame φ, we denote by $\mathrm{dom}(\varphi)$ its *domain* $\mathrm{dom}(\varphi) = \{w_1, \ldots, w_\ell\}$. If $\mathrm{par}(M) \subseteq \mathrm{dom}(\varphi)$, we write $M\varphi$ for the term obtained by replacing each w_i by t_i in M. We note that if in addition M is ground then $t = M\varphi$ is a ground plain term.

3.1 Deducibility, recipes

Classically (see e.g. [Abadi and Cortier 2006]), a ground term t is *deducible* modulo E from an initial frame φ, written $\varphi \vdash_{\mathsf{E}} t$, if there exists $M \in \mathcal{F}_{\mathsf{pub}}[\mathrm{dom}(\varphi)]$ such that $M\varphi =_{\mathsf{E}} t$. This corresponds to the intuition that the attacker may compute (infer) t from φ. For the purpose of our study, we generalize this notion to arbitrary (i.e. non-necessarily initial) frames, and even sets of (non-necessarily ground) deduction facts ϕ, using the notations $\rhd_\phi$ and $\rhd_\phi^{\mathsf{E}}$ defined as follows.

Definition 3.1 *(Deducibility).* Let ϕ be finite set of deductions facts. We say that M *is a recipe of* t *in* ϕ, written $M \rhd_\phi t$, if there exist a (public, ground, non-necessarily linear) n-ary context C and some deduction facts $M_1 \rhd t_1, \ldots, M_n \rhd t_n$ in ϕ such that $M = C[M_1, \ldots, M_n]$ and $t = C[t_1, \ldots, t_n]$. In that case, we say that t is *syntactically deducible* from ϕ, also written $\phi \vdash t$.

We say that M *is a recipe of* t *in* ϕ *modulo* E, written $M \rhd_\phi^{\mathsf{E}} t$, if there exists a term t' such that $M \rhd_\phi t'$ and $t' =_{\mathsf{E}} t$. In that case, we say that t is *deducible from* ϕ *modulo* E, written $\phi \vdash_{\mathsf{E}} t$.

We note that $M \rhd_\varphi t$ is equivalent to $M\varphi = t$ when φ is an initial frame and when t (or equivalently M) is ground. We also note that in the case of a frame φ, since our contexts C are ground and public, $M \rhd_\varphi t$ implies $\mathrm{var}(M, t) = \emptyset$ and $\mathrm{par}(M) \subseteq \mathrm{par}(\varphi)$.

Example 3.2. Consider the equational theory $\mathsf{E}_{\mathsf{enc}}$ described in Example 2.1. Let $\varphi_0 = \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_0, \mathsf{k}), \mathsf{w}_2 \rhd \mathsf{k}\}$ where c_0 is a public constant and k is a private constant. We have that φ_0 is a set of deduction facts. Since, these facts are ground, φ_0 is actually a frame. Moreover, this frame is initial. We have that $\langle \mathsf{w}_2, \mathsf{w}_2 \rangle \rhd_{\varphi_0} \langle \mathsf{k}, \mathsf{k} \rangle$, $\mathsf{c}_0 \rhd_{\varphi_0} \mathsf{c}_0$, and $\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \rhd_{\varphi_0}^{\mathsf{E}_{\mathsf{enc}}} \mathsf{c}_0$.

The frame $\varphi_0^+ = \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_0, \mathsf{k}), \mathsf{w}_2 \rhd \mathsf{k}, \mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \rhd \mathsf{c}_0\}$ is a non-initial frame because of the deduction fact $\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \rhd \mathsf{c}_0$. This deduction fact is actually a consequence of the others, it will be inferred by our algorithm.

3.2 Static equivalence, visible equations

Deducibility does not always suffice for expressing the knowledge of an attacker. In particular, it does not account for the partial information that an attacker may obtain about secrets. 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. This issue motivates the study of visible equations and static equivalence (see [Abadi and Fournet 2001]), defined as follows.

Definition 3.3 *(Static equivalence).* Let φ be an initial frame. The set of *visible equations of* φ *modulo* E is defined as

$$\mathrm{eq}_{\mathsf{E}}(\varphi) = \{M \bowtie N \mid M, N \in \mathcal{F}_{\mathsf{pub}}[\mathrm{dom}(\varphi)],\ M\varphi =_{\mathsf{E}} N\varphi\}$$

where $\bowtie$ is a dedicated commutative symbol. Two initial frames φ_1 and φ_2 with the same domain are *statically equivalent* modulo E, written $\varphi_1 \approx_{\mathsf{E}} \varphi_2$, if their sets of visible equations are equal, i.e. $\mathrm{eq}_{\mathsf{E}}(\varphi_1) = \mathrm{eq}_{\mathsf{E}}(\varphi_2)$.

This definition is in line with static equivalence in the applied pi calculus [Abadi and Fournet 2001] where bound names would be replaced by free private constants.

Example 3.4. Consider again the equational theory $\mathsf{E}_{\mathsf{enc}}$ given in Example 2.1. Let $\varphi_0 = \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_0, \mathsf{k}),\ \mathsf{w}_2 \rhd \mathsf{k}\}$ and $\varphi_1 = \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_1, \mathsf{k}),\ \mathsf{w}_2 \rhd \mathsf{k}\}$ where c_0, c_1 are public constants and k is a private constant. We have that:

—$(\mathsf{enc}(\mathsf{c}_0, \mathsf{w}_2) \bowtie \mathsf{w}_1) \in \mathrm{eq}_{\mathsf{E}_{\mathsf{enc}}}(\varphi_0)$, and
—$(\mathsf{enc}(\mathsf{c}_0, \mathsf{w}_2) \bowtie \mathsf{w}_1) \notin \mathrm{eq}_{\mathsf{E}_{\mathsf{enc}}}(\varphi_1)$.

Hence, $\mathrm{eq}_{\mathsf{E}_{\mathsf{enc}}}(\varphi_0) \neq \mathrm{eq}_{\mathsf{E}_{\mathsf{enc}}}(\varphi_1)$ and the two frames φ_0 and φ_1 are not statically equivalent. However, it can be shown that $\{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_0, \mathsf{k})\} \approx_{\mathsf{E}_{\mathsf{enc}}} \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_1, \mathsf{k})\}$.

For the purpose of finitely describing the set of visible equations $\mathrm{eq}_{\mathsf{E}}(\varphi)$ of an initial frame, we introduce *quantified equations* of the form $\forall z_1, \ldots, z_q.M \bowtie N$ where $z_1, \ldots, z_q \in \mathcal{X}$, $q \geq 0$ and $\mathrm{var}(M, N) \subseteq \{z_1, \ldots, z_q\}$. In what follows, finite sets of quantified equations are denoted Ψ, Ψ_0,... We write $\Psi \models M \bowtie N$ when the ground equation $M \bowtie N$ is a consequence of Ψ in the usual, first-order logic with equality axioms for the relation $\bowtie$ (that is, reflexivity, symmetry, transitivity and compatibility with symbols in $\mathcal{F}_{\mathsf{pub}}$). When no confusion arises, we may refer to quantified equations simply as *equations*. As usual, quantified equations are considered up to renaming of bound variables.

Example 3.5. Consider the equational theory $\mathsf{E}_{\mathsf{hom}}$ given in Example 2.1. Let $\varphi = \{\mathsf{w}_1 \rhd \mathsf{enc}(\langle \mathsf{c}_0, \mathsf{c}_1 \rangle, \mathsf{k}),\ \mathsf{w}_2 \rhd \langle \mathsf{enc}(\mathsf{c}_0, \mathsf{k}), \mathsf{enc}(\mathsf{c}_1, \mathsf{k}) \rangle,\ \mathsf{w}_3 \rhd \mathsf{k}\}$ where c_0 and c_1 are public constants and k is a private constant. In the set $\mathrm{eq}_{\mathsf{E}_{\mathsf{hom}}}(\varphi)$, we have, among others, $\mathsf{w}_1 \bowtie \mathsf{w}_2$ and $\mathsf{dec}(\mathsf{w}_1, M) \bowtie \langle \mathsf{dec}(\mathsf{proj}_1(\mathsf{w}_1), M), \mathsf{dec}(\mathsf{proj}_2(\mathsf{w}_1), M) \rangle$ for every term $M \in \mathcal{F}_{\mathsf{pub}}[\mathrm{dom}(\varphi)]$. Indeed, we have that:

$$\begin{aligned} \mathsf{dec}(\mathsf{w}_1, M)\varphi &= \mathsf{dec}(\mathsf{enc}(\langle \mathsf{c}_0, \mathsf{c}_1 \rangle, \mathsf{k}), M\varphi) \\ &=_{\mathsf{E}_{\mathsf{hom}}} \langle \mathsf{dec}(\mathsf{enc}(\mathsf{c}_0, \mathsf{k}), M\varphi), \mathsf{dec}(\mathsf{enc}(\mathsf{c}_1, \mathsf{k}), M\varphi) \rangle \\ &=_{\mathsf{E}_{\mathsf{hom}}} \langle \mathsf{dec}(\mathsf{proj}_1(\mathsf{w}_1), M), \mathsf{dec}(\mathsf{proj}_2(\mathsf{w}_1), M) \rangle \varphi \end{aligned}$$

This infinite set will be represented with the quantified equation:

$$\forall z.\ \mathsf{dec}(\mathsf{w}_1, z) \bowtie \langle \mathsf{dec}(\mathsf{proj}_1(\mathsf{w}_1), z), \mathsf{dec}(\mathsf{proj}_2(\mathsf{w}_1), z) \rangle.$$

4. MAIN PROCEDURE

In this section, we describe our algorithms for checking deducibility and static equivalence on convergent rewrite systems. After some additional notations, we present the core of the procedure, which consists of a set of transformation rules used to saturate a frame and a finite set of quantified equations. The result of the saturation can be seen as a finite description of the deducible terms and visible equations of the initial frame under consideration. We then show how to use this procedure to decide deducibility and static equivalence, provided that saturation succeeds. (Recall that static equivalence and deduction are undecidable for convergent theories [Abadi and Cortier 2006].)

Soundness and completeness of the saturation procedure are detailed in Section 5. We provide sufficient conditions on the rewrite systems to ensure success of saturation and termination in Section 6 and Section 7.

4.1 Decompositions of rewrite rules

Before stating the procedure, we introduce the following notion of *decomposition* to account for the possible superpositions of an attacker's context (that is, a recipe in our setting) with a left-hand side of rewrite rule.

Definition 4.1 *Decomposition.* Let n, p, q be non-negative integers. A (n, p, q)-*decomposition* of a term l (and by an extension of any rewrite rule $l \to r$) is a (public, ground, non-necessarily linear) context $D \in \mathcal{F}_{\mathsf{pub}}[\mathcal{W}]$ such that $\mathrm{par}(D) = \{\mathsf{w}_1, \ldots, \mathsf{w}_{n+p+q}\}$ and $l = D[l_1, \ldots, l_n, y_1, \ldots, y_p, z_1, \ldots, z_q]$ where

—$l_1, \ldots, l_n$ are mutually-distinct non-variable terms,
—$y_1, \ldots, y_p$ and $z_1, \ldots, z_q$ are mutually-distinct variables, and
—$y_1, \ldots, y_p \in \mathrm{var}(l_1, \ldots, l_n)$ whereas $z_1, \ldots, z_q \notin \mathrm{var}(l_1, \ldots, l_n)$.

A decomposition D is *proper* if it is not a parameter (i.e. $D \neq \mathsf{w}_1$).

In order to avoid unnecessary computations, (n, p, q)-decompositions are considered up to permutations of parameters in the sets $\{\mathsf{w}_1, \ldots, \mathsf{w}_n\}$, $\{\mathsf{w}_{n+1}, \ldots, \mathsf{w}_{n+p}\}$ and $\{\mathsf{w}_{n+p+1}, \ldots, \mathsf{w}_{n+p+q}\}$ respectively.

Example 4.2. Consider the rewrite rule $\mathsf{dec}(\mathsf{enc}(x, y), y) \to x$. This rule admits two proper decompositions up to permutation of parameters:

—$D_1 = \mathsf{dec}(\mathsf{enc}(\mathsf{w}_1, \mathsf{w}_2), \mathsf{w}_2)$ where $n = 0$, $p = 0$, $q = 2$, $z_1 = x$, $z_2 = y$;
—$D_2 = \mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2)$ where $n = 1$, $p = 1$, $q = 0$, $l_1 = \mathsf{enc}(x, y)$ and $y_1 = y$.

Now, consider the rewrite rule $\mathsf{dec}(\langle x, y\rangle, z) \to \langle \mathsf{dec}(x, z), \mathsf{dec}(y, z)\rangle$. This rule also admits two proper decompositions:

—$D_3 = \mathsf{dec}(\langle \mathsf{w}_1, \mathsf{w}_2\rangle, \mathsf{w}_3)$ where $n = 0$, $p = 0$, $q = 3$, $z_1 = x$, $z_2 = y$, $z_3 = z$;
—$D_4 = \mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2)$ where $n = 1$, $p = 0$, $q = 1$, $l_1 = \langle x, y\rangle$, $z_1 = z$.

4.2 Transformation rules

To check deducibility and static equivalence, we proceed by saturating an initial frame, adding some deduction facts and equations satisfied by the frame. We

A. Inferring deduction facts and equations by context reduction

Assume that

$$l = D[l_1, \ldots, l_n, y_1, \ldots, y_p, z_1, \ldots, z_q] \text{ is a proper decomposition of } (l \to r) \in \mathcal{R}$$
$$M_1 \rhd t_1, \ldots, M_{n+p} \rhd t_{n+p} \in \Phi$$
$$(l_1, \ldots, l_n, y_1, \ldots, y_p)\,\sigma = (t_1, \ldots, t_{n+p})$$

(1) If there exists $M = \mathrm{Ctx}(\Phi \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\} \vdash^{?}_{\mathcal{R}} r\sigma)$ (with $M \neq \bot$), then

$$(\Phi, \Psi) \Longrightarrow (\Phi, \Psi \cup \{\forall z_1, \ldots, z_q.D[M_1, \ldots, M_{n+p}, z_1 \ldots, z_q] \bowtie M\}) \qquad (\mathbf{A.1})$$

(2) Else, if $(r\sigma)\!\downarrow_{\mathcal{R}}$ is ground, then

$$(\Phi, \Psi) \Longrightarrow (\Phi \cup \{M_0 \rhd (r\sigma)\!\downarrow_{\mathcal{R}}\}, \Psi \cup \{\forall z_1, \ldots, z_q.D[M_1, \ldots, M_{n+p}, z_1 \ldots, z_q] \bowtie M_0\}) \qquad (\mathbf{A.2})$$

where $M_0 = D[M_1, \ldots, M_{n+p}, \mathsf{a}, \ldots, \mathsf{a}]$ for some fixed public constant a.

(3) Otherwise, $(\Phi, \Psi) \Longrightarrow \bot$ $(\mathbf{A.3})$

B. Inferring deduction facts and equations syntactically

Assume that $M_0 \rhd t_0, \ldots, M_n \rhd t_n \in \Phi \qquad t = f(t_1, \ldots, t_n) \in \mathrm{st}(t_0) \qquad f \in \mathcal{F}_{\mathsf{pub}}$

(1) If there exists M such that $(M \rhd t) \in \Phi$,

$$(\Phi, \Psi) \Longrightarrow (\Phi, \Psi \cup \{f(M_1, \ldots, M_n) \bowtie M\}) \qquad (\mathbf{B.1})$$

(2) Otherwise, $(\Phi, \Psi) \Longrightarrow (\Phi \cup \{f(M_1, \ldots, M_n) \rhd t\}, \Psi)$ $(\mathbf{B.2})$

Fig. 1. Transformation rules

consider *states* that are either the failure state $\bot$ or a pair (Φ, Ψ) formed by a one-to-one frame Φ in $\mathcal{R}$-reduced form and a finite set of quantified equations Ψ.

Given an initial frame φ, our procedure starts from an initial state associated to φ, denoted by $\mathrm{Init}(\varphi)$, obtained by reducing φ and replacing duplicated terms by equations. Formally, $\mathrm{Init}(\varphi)$ is the result of a procedure recursively defined as follows: $\mathrm{Init}(\emptyset) = (\emptyset, \emptyset)$, and assuming $\mathrm{Init}(\varphi) = (\Phi, \Psi)$, we have

$$\mathrm{Init}(\varphi \uplus \{w \rhd t\}) = \begin{cases} (\Phi, \Psi \cup \{w \bowtie w'\}) & \text{if there exists some } w' \rhd t\!\downarrow_{\mathcal{R}} \in \Phi \\ (\Phi \cup \{w \rhd t\!\downarrow_{\mathcal{R}}\}, \Psi) & \text{otherwise.} \end{cases}$$

Example 4.3. Consider the frames φ_0, φ_1 and φ introduced respectively in Example 3.4 and Example 3.5. We have that $\mathrm{Init}(\varphi_0) = (\varphi_0, \emptyset)$, $\mathrm{Init}(\varphi_1) = (\varphi_1, \emptyset)$ and $\mathrm{Init}(\varphi) = (\{\mathsf{w}_1 \rhd \langle \mathsf{enc}(\mathsf{c}_0, \mathsf{k}), \mathsf{enc}(\mathsf{c}_1, \mathsf{k})\rangle, \mathsf{w}_3 \rhd \mathsf{k}\},\ \{\mathsf{w}_1 \bowtie \mathsf{w}_2\})$.

The main part of our procedure consists in saturating a state (Φ, Ψ) by means of the transformation rules described in Figure 1. Intuitively, the **A** rules allow us to get rid of the equational theory, still ensuring the completeness of our procedure. More precisely, the **A** rules are designed for applying a rewrite step $l \to r$ on top of $D[t_1, \ldots, t_{n+p}, z_1, \ldots, z_q]$ where D is a public context and $t_1, \ldots, t_{n+p}$ are deducible terms already in Φ. Depending on the resulting term $(r\sigma)\!\downarrow_{\mathcal{R}}$, i.e. the one obtained after application of the rewrite rule $l \to r$, we obtain an instance of **A.1**, **A.2**, or **A.3**. If the resulting term $(r\sigma)\!\downarrow_{\mathcal{R}}$ is already deducible (in some specific sense that we make precise below) then a corresponding equation is added (rule **A.1**); or else if it is ground, the corresponding deduction fact is added to the state (rule **A.2**); otherwise, the procedure may fail (rule **A.3**). Note that, in case of an application of the rule **A.2**, the remaining variables are replaced by an arbitrary

public constant a in order to obtain a ground recipe M. The **B** rules do not take into account the underlying equational theory. They are meant to add syntactically deducible subterms (rule **B.2**) or related equations (rule **B.1**) when the subterm is already in Φ.

For technical reasons, rule **A.1** is parametrized by a function Ctx that outputs either a recipe M or the special symbol $\bot$. This function has to satisfy the following properties:

(a) if $\phi \vdash t{\downarrow_{\mathcal{R}}}$, then $\mathrm{Ctx}(\phi \vdash^?_{\mathcal{R}} t) \neq \bot$;
(b) if $M = \mathrm{Ctx}(\phi \vdash^?_{\mathcal{R}} t)$ then there exists s such that $M \rhd_\phi s$ and $t \to^*_{\mathcal{R}} s$. (This justifies the notation $\phi \vdash^?_{\mathcal{R}} t$ used to denote a specific deducibility problem.)

Property (a) ensures that the rules transform a state into a state (and more precisely that the resulting frame in **(A.2)** is still one-to-one). Property (b) guarantees the soundness of the new equation in **(A.1)**. Requiring $t \to^*_{\mathcal{R}} s$ instead of $t =_{\mathsf{E}} s$ is necessary for the proof of completeness. In what follows, a *function* Ctx is any function satisfying the two properties (a) and (b).

A simple choice for $\mathrm{Ctx}(\phi \vdash^?_{\mathcal{R}} t)$ is to solve the deducibility problem $\phi \vdash^? t{\downarrow_{\mathcal{R}}}$ in the empty equational theory, and then return a corresponding recipe M, if any. (This problem is easily solved by induction on $t{\downarrow_{\mathcal{R}}}$.) We will see in Section 6 that this choice is sufficient to avoid failure for a large class of equational theories, namely the class of layered convergent theories. However the proof of this fact relies on an intermediate result that uses a different choice of Ctx.

Example 4.4. Consider the frame φ_0 previously described in Example 3.4 and an arbitrary function Ctx. First, we apply rule **B.2** with $t = \mathsf{c}_0$, and $t_0 = \mathsf{enc}(\mathsf{c}_0, \mathsf{k})$. This leads us to add the deduction fact $\mathsf{c}_0 \rhd \mathsf{c}_0$. Let $\Phi_0 = \varphi_0 \cup \{\mathsf{c}_0 \rhd \mathsf{c}_0\}$. Then, we can apply rule **A.1** as follows. Consider the rewrite rule $\mathsf{dec}(\mathsf{enc}(x, y), y) \to x$, the decomposition D_2 given in Example 4.2 with $l_1\sigma = t_1 = \mathsf{enc}(\mathsf{c}_0, \mathsf{k})$, and $y_1\sigma = \mathsf{k}$. We have that $\mathrm{Ctx}(\varphi_0 \vdash^?_{\mathcal{R}} \mathsf{c}_0) = \mathsf{c}_0$. Hence, we have that:

$$\mathrm{Init}(\varphi_0) = (\varphi_0, \emptyset) \Longrightarrow (\Phi_0, \emptyset) \Longrightarrow (\Phi_0, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_0\}).$$

In other words, since we know the key k through w_2, we can check that the decryption of w_1 by w_2 leads to the public constant c_0. Next we can apply rule **B.1** with $t_0 = t = \mathsf{enc}(\mathsf{c}_0, \mathsf{k})$, $t_1 = \mathsf{c}_0$, $t_2 = \mathsf{k}$ and $f = \mathsf{enc}$. This gives us:

$$(\Phi_0, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_0\}) \Longrightarrow (\Phi_0, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_0, \mathsf{enc}(\mathsf{c}_0, \mathsf{w}_2) \bowtie \mathsf{w}_1\}).$$

Lastly, we can apply **A.1** with the decomposition D_1 given in Example 4.2. We have that $\mathrm{Ctx}(\Phi_0 \cup \{z_1 \rhd z_1, z_2 \rhd z_2\} \vdash^?_{\mathcal{R}} \mathsf{c}_0) = z_1$. Hence, we reach the following state: $(\Phi_0, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_0, \mathsf{enc}(\mathsf{c}_0, \mathsf{w}_2) \bowtie \mathsf{w}_1, \forall z_1, z_2.\mathsf{dec}(\mathsf{enc}(z_1, z_2), z_2) \bowtie z_1\})$. No more rules can then modify the state.

Similarly for φ_1, let $\Phi_1 = \varphi_1 \cup \{\mathsf{c}_1 \rhd \mathsf{c}_1\}$, we obtain that:

$$\begin{aligned}
&\mathrm{Init}(\varphi_1) = (\varphi_1, \emptyset)\\
&\quad\Longrightarrow (\Phi_1, \emptyset)\\
&\quad\Longrightarrow (\Phi_1, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_1\})\\
&\quad\Longrightarrow (\Phi_1, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_1, \mathsf{enc}(\mathsf{c}_1, \mathsf{w}_2) \bowtie \mathsf{w}_1\})\\
&\quad\Longrightarrow (\Phi_1, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_1, \mathsf{enc}(\mathsf{c}_1, \mathsf{w}_2) \bowtie \mathsf{w}_1, \forall z_1, z_2.\mathsf{dec}(\mathsf{enc}(z_1, z_2), z_2) \bowtie z_1\}).
\end{aligned}$$

Example 4.5. Consider the frame φ described in Example 3.5 and an arbitrary function Ctx. We can choose to apply the rule **A**. Consider the rewrite rule $\mathsf{dec}(\langle x, y\rangle, z) \to \langle \mathsf{dec}(x, z), \mathsf{dec}(y, z)\rangle$, the decomposition D_4 given in Example 4.2 and $t_1 = \langle \mathsf{enc}(\mathsf{c}_0, \mathsf{k}), \mathsf{enc}(\mathsf{c}_1, \mathsf{k})\rangle$. We have that

$$r\sigma = \langle \mathsf{dec}(\mathsf{enc}(\mathsf{c}_0, \mathsf{k}), z_1), \mathsf{dec}(\mathsf{enc}(\mathsf{c}_1, \mathsf{k}), z_1)\rangle = r\sigma{\downarrow_{\mathcal{R}}}.$$

The condition required in case (1) is not fulfilled and the condition stated in case (2) is false. Hence, we have that $\mathrm{Init}(\varphi) \Longrightarrow \bot$.

However, note that another strategy of rules application allows us to consider this decomposition. For this, it is sufficient to apply first **A.2** rules to add the deduction facts $\mathsf{proj}_1(\mathsf{w}_2) \rhd \mathsf{enc}(\mathsf{c}_0, \mathsf{k})$ and $\mathsf{proj}_2(\mathsf{w}_2) \rhd \mathsf{enc}(\mathsf{c}_1, \mathsf{k})$. Now, we have that $r\sigma{\downarrow_{\mathcal{R}}}$ is syntactically deducible: the condition required in case (1) is fulfilled and we finally add the equation: $\forall z_1.\mathsf{dec}(\mathsf{w}_1, z_1) \bowtie \langle \mathsf{dec}(\mathsf{proj}_1(\mathsf{w}_2), z_1), \mathsf{dec}(\mathsf{proj}_2(\mathsf{w}_2), z_1)\rangle$.

We write $\Longrightarrow^*$ for the transitive and reflexive closure of $\Longrightarrow$. The definitions of Ctx and of the transformation rules ensure that whenever $S \Longrightarrow^* S'$ and S is a state, then S' is also a state, with the same parameters unless $S' = \bot$.

4.3 Main theorem

We now state the soundness and the completeness of the transformation rules provided that a *saturated state* is reached, that is, a state $S \neq \bot$ such that $S \Longrightarrow S'$ implies $S' = S$. The technical lemmas involved in the proof of this theorem are detailed in Section 5.

Theorem 4.6 (soundness and completeness). *Let* E *be an equational theory generated by a convergent rewrite system* $\mathcal{R}$*. Let* φ *be an initial frame and* (Φ, Ψ) *be a saturated state such that* $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$.

(1) *For all* $M \in \mathcal{F}_{\mathsf{pub}}[\mathrm{par}(\varphi)]$ *and* $t \in \mathcal{F}[\emptyset]$*, we have that:*

$$M\varphi =_{\mathsf{E}} t \quad \Leftrightarrow \quad \exists N \text{ such that } \Psi \models M \bowtie N \text{ and } N \rhd_\Phi t{\downarrow_{\mathcal{R}}}.$$

(2) *For all* M*,* $N \in \mathcal{F}_{\mathsf{pub}}[\mathrm{par}(\varphi) \cup \mathcal{X}]$*, we have that:*

$$M\varphi =_{\mathsf{E}} N\varphi \Leftrightarrow \Psi \models M \bowtie N.$$

We note that this theorem applies to any saturated state reachable from the initial frame. Moreover, while the saturation procedure is sound and complete, it may not terminate, or it may *fail* if rule **A.3** becomes the only applicable rule at some point of computation. In Section 6 and Section 7, we explore several sufficient conditions to prevent failure and ensure termination.

4.4 Application to deduction and static equivalence

Decision procedures for deduction and static equivalence modulo E follow from Theorem 4.6.

Algorithm for deduction. Let φ be an initial frame and t be a ground term. The procedure for checking $\varphi \vdash_{\mathsf{E}} t$ runs as follows:

(1) Apply the transformation rules to obtain (if any) a saturated state (Φ, Ψ) such that $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$;

(2) Return *yes* if there exists N such that $N \rhd_\Phi t{\downarrow_\mathcal{R}}$ (that is, the $\mathcal{R}$-reduced form of t is syntactically deducible from Φ); otherwise return *no*.

PROOF. If the algorithm returns *yes*, this means that there exists N such that $N \rhd_\Phi t{\downarrow_\mathcal{R}}$. Thanks to Theorem 4.6 (1), we have that $N\varphi =_\mathsf{E} t$, i.e. $N \rhd_\varphi^\mathsf{E} t$.

Conversely, if t is deducible from φ, then there exists M such that $M\varphi =_\mathsf{E} t$. By Theorem 4.6 (1), there exists N such that $N \rhd_\Phi t{\downarrow_\mathcal{R}}$. The algorithm returns *yes*. □

Example 4.7. Consider the frame $\varphi_0 = \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_0, \mathsf{k}), \mathsf{w}_2 \rhd \mathsf{k}\}$ introduced in Example 3.2 and let $t_1 = \langle \mathsf{k}, \mathsf{k} \rangle$ and $t_2 = \mathsf{c}_0$. Let (Φ_0, Ψ_0) be the saturated state described in Example 4.4. We have that:

$$(\Phi_0, \Psi_0) = (\varphi_0, \{\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_0, \mathsf{enc}(\mathsf{c}_0, \mathsf{w}_2) \bowtie \mathsf{w}_1\}).$$

Then, it is easy to see that our algorithm for deduction will return *yes* for both terms t_1 and t_2. Indeed, those terms are syntactically deducible from φ_0.

Algorithm for static equivalence. Let φ_1 and φ_2 be two initial frames. The procedure for checking $\varphi_1 \approx_\mathsf{E} \varphi_2$ runs as follows:

(1) Apply the transformation rules to obtain (if possible) two saturated states (Φ_1, Ψ_1) and (Φ_2, Ψ_2) such that $\mathrm{Init}(\varphi_i) \Longrightarrow^* (\Phi_i, \Psi_i)$, $i = 1, 2$;
(2) For $\{i, j\} = \{1, 2\}$, for every equation $(\forall z_1, \ldots, z_\ell . M \bowtie N)$ in Ψ_i, check that $M\varphi_j =_\mathsf{E} N\varphi_j$ — that is, in other words, $(M\varphi_j){\downarrow_\mathcal{R}} = (N\varphi_j){\downarrow_\mathcal{R}}$;
(3) If so return *yes*; otherwise return *no*.

PROOF. If the algorithm returns *yes*, this means that $M\varphi_2 =_\mathsf{E} N\varphi_2$ for every equation $(\forall z_1, \ldots, z_\ell . M \bowtie N)$ in Ψ_1. Let $M \bowtie N \in \mathrm{eq}_\mathsf{E}(\varphi_1)$. By definition of $\mathrm{eq}_\mathsf{E}(\varphi_1)$, we have that $M\varphi_1 =_\mathsf{E} N\varphi_1$. Thanks to Theorem 4.6 (2), we have that $\Psi_1 \models M \bowtie N$. As all the equations in Ψ_1 are satisfied by φ_2 modulo E, we deduce that $M\varphi_2 =_\mathsf{E} N\varphi_2$, i.e. $M \bowtie N \in \mathrm{eq}(\varphi_2)$. The other inclusion, $\mathrm{eq}_\mathsf{E}(\varphi_2) \subseteq \mathrm{eq}_\mathsf{E}(\varphi_1)$, is proved in the same way.

Conversely, assume now that $\varphi_1 \approx_\mathsf{E} \varphi_2$, i.e. $\mathrm{eq}_\mathsf{E}(\varphi_1) = \mathrm{eq}_\mathsf{E}(\varphi_2)$. Consider a quantified equation $\forall z_1, \ldots, z_\ell . M \bowtie N$ in Ψ_1 and let us show that $M\varphi_2 =_\mathsf{E} N\varphi_2$. (The other case is done in a similar way, and we will conclude that the algorithm returns *yes*.) Let $\mathsf{c}_1, \ldots, \mathsf{c}_\ell$ be free public constants not occurring in M and N, and let $(M', N') = (M, N)\{z_1 \mapsto \mathsf{c}_1, \ldots, z_\ell \mapsto \mathsf{c}_\ell\}$. Since $\Psi_1 \models M' \bowtie N'$, by Theorem 4.6 (2), we have that $M'\varphi_1 =_\mathsf{E} N'\varphi_1$. Besides, M' and N' are ground and $\mathrm{par}(M', N') \subseteq \mathrm{par}(\Psi_1) \subseteq \mathrm{par}(\varphi_1)$. Thus, $(M' \bowtie N') \in \mathrm{eq}_\mathsf{E}(\varphi_1) \subseteq \mathrm{eq}_\mathsf{E}(\varphi_2)$ and $M'\varphi_2 =_\mathsf{E} N'\varphi_2$. As the constants $\mathsf{c}_1, \ldots, \mathsf{c}_\ell$ are free in E and do not occur in M and N, by replacement, we obtain that $M\varphi_2 =_\mathsf{E} N\varphi_2$. □

Example 4.8. Consider the frames $\varphi_i = \{\mathsf{w}_1 \rhd \mathsf{enc}(\mathsf{c}_i, \mathsf{k}), \mathsf{w}_2 \rhd \mathsf{k}\}$ introduced in Example 3.4. Let (Φ_0, Ψ_0) and (Φ_1, Ψ_1) be the two saturated states described in Example 4.4. We have that $\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2) \bowtie \mathsf{c}_0 \in \Psi_0$, and

$$\mathsf{dec}(\mathsf{w}_1, \mathsf{w}_2)\varphi_1 =_{\mathsf{E}_{\mathsf{enc}}} \mathsf{c}_1 \neq_{\mathsf{E}_{\mathsf{enc}}} \mathsf{c}_0 = \mathsf{c}_0\varphi_1.$$

Hence, our algorithm returns *no*. The two frames φ_0 and φ_1 are not statically equivalent.

5. SOUNDNESS AND COMPLETENESS OF THE SATURATION

The goal of this section is to prove Theorem 4.6. Section 5.1 is devoted to establish soundness of our saturation procedure, i.e. the $\Leftarrow$ direction of Theorem 4.6. Showing the other direction, i.e. completeness, is more involved and is detailed in Section 5.2.

5.1 Soundness

First, the transformation rules are sound in the sense that, along the saturation process, we add only deducible terms and valid equations with respect to the initial frame.

LEMMA 5.1 (SOUNDNESS). *Let φ be an initial frame and (Φ, Ψ) be a state such that* $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$. *Then, we have that*

(1) $M \rhd_\Phi t \;\Rightarrow\; M\varphi =_{\mathsf{E}} t$ *for all* $M \in \mathcal{F}_{\mathsf{pub}}[\mathrm{dom}(\varphi)]$ *and* $t \in \mathcal{F}[\emptyset]$;
(2) $\Psi \models M \bowtie N \;\Rightarrow\; M\varphi =_{\mathsf{E}} N\varphi$ *for all* $M, N \in \mathcal{F}_{\mathsf{pub}}[\mathrm{dom}(\varphi) \cup \mathcal{X}]$.

PROOF. We prove this result by induction on the derivation $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$. To prove the induction step, we perform a case analysis on the inference rule. The case of the **B** rules is quite straightforward. For the **A** rule, we have to rely on the definition of the function Ctx (in case of **A.1**) and we rely on the fact that E is stable by replacement of variables with constants to establish soundness of **A.2**. The soundness of the rule **A.3** is trivial. More formally, we have that:

Base case: We have that $(\Phi, \Psi) = \mathrm{Init}(\varphi)$ and we easily conclude.

Induction case: In such a case, we have $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi', \Psi') \Longrightarrow (\Phi, \Psi)$.

Let us first notice two facts.

(1) Let M and t be such that $M \rhd_\Phi t$. By definition of $\rhd_\Phi$, there exist a public context C and some deduction facts $M'_1 \rhd t'_1, \ldots, M'_n \rhd t'_n \in \Phi$ such that $M = C[M'_1, \ldots M'_n]$ and $t = C[t'_1, \ldots, t'_n]$. In order to prove that $M\varphi =_{\mathsf{E}} t$, it is sufficient to show that $M' \rhd^{\mathsf{E}}_{\varphi} t'$ for every $M' \rhd t' \in \Phi$. By induction hypothesis, this holds for the deduction facts in Φ', thus it remains to show that $M' \rhd^{\mathsf{E}}_{\varphi} t'$ for every fact $M' \rhd t' \in \Phi - \Phi'$.
(2) Let M, N be two terms such that $\Psi \models M \bowtie N$. To establish that $M\varphi =_{\mathsf{E}} N\varphi$, it is sufficient to prove that $M'\varphi =_{\mathsf{E}} N'\varphi$ for every $(\forall z_1, \ldots, z_q.M' \bowtie N') \in \Psi$. By induction hypothesis, this holds for the equations in Ψ', thus it remains to show that $M'\varphi =_{\mathsf{E}} N'\varphi$ for every equation $(\forall z_1, \ldots, z_q.M' \bowtie N') \in \Psi - \Psi'$.

Next we perform a case analysis on the inference rule used in $(\Phi', \Psi') \Longrightarrow (\Phi, \Psi)$.

First, consider the case of rule **A**. Let $l \to r \in \mathcal{R}$ be the rewrite rule, D the decomposition, and $M_1 \rhd t_1, \ldots, M_{n+p} \rhd t_{n+p}$ the facts involved in this step.

*Rule **A.2***: We need to show that

—$D[M_1, \ldots, M_{n+p}, \mathsf{a}, \ldots, \mathsf{a}]\varphi =_{\mathsf{E}} (r\sigma)\downarrow_{\mathcal{R}}$, and
—$D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q]\varphi =_{\mathsf{E}} D[M_1, \ldots, M_{n+p}, \mathsf{a}, \ldots, \mathsf{a}]\varphi$.

We note that $D[t_1, \ldots, t_{n+p}, z_1, \ldots, z_q] = l\sigma \to r\sigma \to^* (r\sigma)\downarrow_{\mathcal{R}}$. Besides, by induction hypothesis we have that $M_i\varphi =_{\mathsf{E}} t_i$ for $1 \le i \le n + p$. Given that $(r\sigma)\downarrow_{\mathcal{R}}$

is ground, and applying the substitution $\{z_1 \mapsto \mathsf{a}, \ldots, z_q \mapsto \mathsf{a}\}$ to the equation $D[t_1, \ldots, t_{n+p}, z_1, \ldots, z_q] =_\mathsf{E} (r\sigma)\downarrow_\mathcal{R}$, we obtain:

$$\begin{array}{rcl} D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q]\varphi & =_\mathsf{E} & D[t_1, \ldots, t_{n+p}, z_1, \ldots, z_q] \\ & =_\mathsf{E} & (r\sigma)\downarrow_\mathcal{R} \\ & =_\mathsf{E} & D[t_1, \ldots, t_{n+p}, \mathsf{a}, \ldots, \mathsf{a}] \\ & =_\mathsf{E} & D[M_1, \ldots, M_{n+p}, \mathsf{a}, \ldots, \mathsf{a}]\varphi \end{array}$$

Rule ***A.1***: We need to show $D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q]\varphi =_\mathsf{E} M\varphi$. As before, we have $D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q]\varphi =_\mathsf{E} (r\sigma)\downarrow_\mathcal{R}$. We also know that there exists s such that $M \rhd_{\Phi'^+} s$ and $r\sigma \to_\mathcal{R}^* s$ where $\Phi'^+ = \Phi' \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\}$ thanks to property (b) of Ctx. Let θ be the substitution $\{z_1 \mapsto \mathsf{a}_1, \ldots, z_q \mapsto \mathsf{a}_q\}$ where $\mathsf{a}_1, \ldots, \mathsf{a}_q$ are public constants that do not occur in Φ', M, and s. We have that $M\theta \rhd_{\Phi'} s\theta$. Hence, using the induction hypothesis, we have that $(M\theta)\varphi =_\mathsf{E} s\theta$ thus $M\varphi =_\mathsf{E} s$, *i.e.* $M\varphi =_\mathsf{E} (r\sigma)\downarrow_\mathcal{R}$. This allows us to conclude.

Rule ***A.3***: In such a case, the result trivially holds.

Second, we consider the case of **B** rules. Let $t = f(t_1, \ldots, t_n) \in \mathrm{st}(t_0)$, $f \in \mathcal{F}_\mathsf{pub}$ and $M_0 \rhd t_0, \ldots, M_n \rhd t_n \in \Phi$ be involved in the step $(\Phi', \Psi') \Longrightarrow (\Phi, \Psi)$.

Rule ***B.1***: By induction hypothesis, $M_i\varphi =_\mathsf{E} t_i$ for every $1 \leq i \leq n$ and $M\varphi =_\mathsf{E} t$, hence $f(M_1, \ldots, M_n)\varphi =_\mathsf{E} f(t_1, \ldots, t_n) = t =_\mathsf{E} M\varphi$.

Rule ***B.2***: By induction hypothesis, $M_i\varphi =_\mathsf{E} t_i$ for every $1 \leq i \leq n$, hence $f(M_1, \ldots, M_n)\varphi =_\mathsf{E} f(t_1, \ldots, t_n) = t$. □

5.2 Completeness

The next three lemmas are dedicated to the completeness of **B** rules (Lemma 5.2 and Lemma 5.3) and **A** rules (Lemma 5.4).

Lemma 5.2 ensures that a saturated state (Φ, Ψ) contains all the deduction facts $M \rhd t$ where t is a subterm of Φ that is syntactically deducible, whereas Lemma 5.3 ensures that saturated states account for all the syntactic equations possibly visible on the frame.

Lemma 5.2 (completeness, syntactic deduction). *Let (Φ, Ψ) be a state, $M_0 \rhd t_0 \in \Phi$. Let N, t be two terms such that $t \in \mathrm{st}(t_0)$ and $N \rhd_\Phi t$. Then there exists (Φ', Ψ') and N' such that:*

*—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$ using **B** rules, and*
—$N' \rhd t \in \Phi'$ and $\Psi' \models N \bowtie N'$.

The proof of Lemma 5.2 is postponed to the appendix. It uses a simple induction on the context C witnessing the fact that t is syntactically deducible from Φ.

Lemma 5.3 (completeness, syntactic equations). *Let (Φ, Ψ) be a state, and M, N be two terms such that $M \rhd_\Phi t$ and $N \rhd_\Phi t$ for some term t. Then there exists (Φ', Ψ') such that:*

*—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$ using **B** rules, and*
—$\Psi' \models M \bowtie N$.

PROOF. (sketch) Let C, C' be the contexts witnessing $M \rhd_\Phi t$ and $N \rhd_\Phi t$. Assume that C is smaller than C'. The proof is done by induction on C. When C is reduced to a hole, we apply Lemma 5.2 to conclude. Otherwise, we have that $C = f(C_1, \ldots, C_r)$ and $C' = f(C'_1, \ldots, C'_r)$. We easily conclude by applying our induction hypothesis on C_i, C'_i for each $1 \leq i \leq r$. The detailed proof is presented in appendix A. □

Now, we know that terms that are syntactically deducible from the frame and syntactic equation visible on the frame will be added during our saturation procedure. It remains to take into account the underlying equational theory. This is the purpose of Lemma 5.4 that deals with the reduction of a deducible term along the rewrite system $\mathcal{R}$. Using the fact that $\mathcal{R}$ is convergent, this allows us to prove that every deducible term from a saturated frame is syntactically deducible.

LEMMA 5.4 (COMPLETENESS, CONTEXT REDUCTION). *Let (Φ, Ψ) be a state and M, t, t' be three terms such that $M \rhd_\Phi t$ and $t \to_\mathcal{R} t'$. Then, either $(\Phi, \Psi) \Longrightarrow^* \bot$ or there exist (Φ', Ψ'), M' and t'' such that*

—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$,
—$M' \rhd_{\Phi'} t''$ *with* $t' \to^*_\mathcal{R} t''$, *and*
—$\Psi' \models M \bowtie M'$.

Besides, in both cases, the corresponding derivation from (Φ, Ψ) can be chosen to consist of a number of ***B*** *rules, possibly followed by one instance of* ***A*** *rule involving the same rewrite rule $l \to r$ as the rewrite step $t \to_\mathcal{R} t'$.*

PROOF. (sketch) The detailed proof of Lemma 5.4 is left to the appendix. We describe here its main arguments. Since $t \to_\mathcal{R} t'$, there exist a position α, a substitution σ and a rewrite rule $l \to r \in \mathcal{R}$ such that $t|_\alpha = l\sigma$ and $t' = t[r\sigma]_\alpha$. Let C be a context witnessing the fact that $M \rhd_\Phi t$. Since terms in $\mathrm{im}(\Phi)$ are $\mathcal{R}$-reduced, α is actually a position in C. Thus, the rewriting step mentioned above corresponds to a proper (n, p, q)-decomposition D of l: $l = D[l_1, \ldots, l_n, y_1, \ldots y_p, z_1, \ldots z_q]$. We can show that $M|_\alpha \rhd_\Phi l\sigma$ and $D[M_1, \ldots, M_n, N_1, \ldots, N_{p+q}] \rhd_\Phi l\sigma$ where

—$M_1 \rhd t_1$, …, $M_n \rhd t_n$ are deduction facts in Φ,
—for every $1 \leq j \leq p$, $N_j \rhd_\Phi y_j\sigma$, and
—for every $1 \leq k \leq q$, $N_{p+k} \rhd_\Phi z_k\sigma$.

Thus, by Lemma 5.3, there exists a derivation $(\Phi, \Psi) \Longrightarrow^* (\Phi_1, \Psi_1)$ using **B** rules such that $\Psi_1 \models M|_\alpha \bowtie D[M_1, \ldots, M_n, N_1, \ldots, N_{p+q}]$.

Besides, $y_j\sigma$ is a subterm of some $l_i\sigma = t_i$. Since $N_j \rhd_\Phi y_j\sigma$, by applying Lemma 5.2 repeatedly, we deduce that there exist some term M_{n+1}, …, M_{n+p} and a derivation $(\Phi_1, \Psi_1) \Longrightarrow^* (\Phi_2, \Psi_2)$ using **B** rules such that for all j,

—$M_{n+j} \rhd y_j\sigma$ is in Φ_2, and
—$\Psi_2 \models M_{n+j} \bowtie N_j$.

Let $N = D[M_1, \ldots, M_{n+p}, N_{p+1}, \ldots, N_{p+q}]$. We deduce that $N \rhd_{\Phi_2} l\sigma$, and

$$\Psi_2 \models M|_\alpha \bowtie D[M_1, \ldots, M_n, N_1, \ldots, N_{p+q}] \bowtie N$$

We now consider the application to (Φ_2, Ψ_2) of a **A** rule that involves the rewrite rule $l \to r$, the decomposition D, the plain terms $(t_1, \ldots, t_{n+p}) = (l_1, \ldots, l_n, y_1, \ldots, y_p)\sigma$. Depending on whether $(r\sigma)\downarrow_{\mathcal{R}}$ is ground and $\mathrm{Ctx}(\Phi_2^+ \vdash^?_{\mathcal{R}} r\sigma') = \bot$, we conclude by applying **A.1**, **A.2** or **A.3**. □

5.3 Main theorem

We are now able to prove soundness and completeness of our transformation rules provided that a saturated state is reached.

THEOREM 4.6 (SOUNDNESS AND COMPLETENESS). *Let* E *be an equational theory generated by a convergent rewrite system* $\mathcal{R}$*. Let* φ *be an initial frame and* (Φ, Ψ) *be a saturated state such that* $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$.

(1) For all $M \in \mathcal{F}_{\mathsf{pub}}[\mathrm{par}(\varphi)]$ *and* $t \in \mathcal{F}[\emptyset]$*, we have that:*

$$M\varphi =_{\mathsf{E}} t \quad \Leftrightarrow \quad \exists N \text{ such that } \Psi \models M \bowtie N \text{ and } N \rhd_\Phi t\downarrow_{\mathcal{R}}.$$

(2) For all $M, N \in \mathcal{F}_{\mathsf{pub}}[\mathrm{par}(\varphi) \cup \mathcal{X}]$*, we have that:*

$$M\varphi =_{\mathsf{E}} N\varphi \Leftrightarrow \Psi \models M \bowtie N.$$

PROOF. Let φ be an initial frame and (Φ, Ψ) be a saturated state such that $\mathrm{Init}(\varphi) \Rightarrow^* (\Phi, \Psi)$.

1.($\Leftarrow$) Let M, N and t be such that $\Psi \models M \bowtie N$ and $N \rhd_\Phi t\downarrow_{\mathcal{R}}$ (thus in particular $N \rhd^{\mathsf{E}}_\Phi t$). Thanks to Lemma 5.1, we have that $M\varphi =_{\mathsf{E}} N\varphi =_{\mathsf{E}} t$.

($\Rightarrow$) Let M and t be such that $M\varphi =_{\mathsf{E}} t$. We have that $M \rhd_\Phi t_0 \to^* t\downarrow_{\mathcal{R}}$ for some term t_0. We show the result by induction on t_0 equipped with the order $<$ induced by the rewriting relation ($t < t'$ if and only if $t' \to^+ t$).

Base case: $M \rhd_\Phi t_0 = t\downarrow_{\mathcal{R}}$. Let $N = M$, we have $\Psi \models M \bowtie N$ and $N \rhd_\Phi t\downarrow_{\mathcal{R}}$.

Induction case: $M \rhd_\Phi t_0 \to^+ t\downarrow_{\mathcal{R}}$. Let t' be such that $M \rhd_\Phi t_0 \to t' \to^* t\downarrow_{\mathcal{R}}$. Thanks to Lemma 5.4 and since (Φ, Ψ) is already saturated[1], we deduce that there exist N' and t'' such that $N' \rhd_\Phi t''$, $t' \to^* t''$, and $\Psi \models M \bowtie N'$. We have that $N' \rhd_\Phi t'' \to^* t\downarrow_{\mathcal{R}}$ and $t'' \leq t' < t_0$. Thus, we can apply our induction hypothesis and we obtain that there exists N such that $\Psi \models N' \bowtie N$ and $N \rhd_\Phi t\downarrow_{\mathcal{R}}$.

2.($\Leftarrow$) By Lemma 5.1, $\Psi \models M \bowtie N$ implies $M\varphi =_{\mathsf{E}} N\varphi$.

($\Rightarrow$) Let M and N such that $M\varphi =_{\mathsf{E}} N\varphi$. This means that there exists t such that $M\varphi =_{\mathsf{E}} t$ and $N\varphi =_{\mathsf{E}} t$. By applying 1, we deduce that there exists M', N' such that: $\Psi \models M \bowtie M'$, $M' \rhd_\Phi t\downarrow_{\mathcal{R}}$, $\Psi \models N \bowtie N'$ and $N' \rhd_\Phi t\downarrow_{\mathcal{R}}$. Thanks to Lemma 5.3 and since (Φ, Ψ) is already saturated, we easily deduce that $\Psi \models M' \bowtie N'$, and thus $\Psi \models M \bowtie N$. □

We proved that saturated frames yield sound and complete characterizations of deducible terms and visible equations of their initial frames. Yet, the saturation procedure may still not terminate, or fail due to rule **A.3**.

[1]Note that rule **A.3** is never applicable on a saturated state.

6. NON-FAILURE

As shown by the following example (from [Ciobâcă et al. 2009]), our procedure may fail.

Example 6.1. Consider the theory $\mathsf{E}_{\mathsf{mal}}$ given below:

$$\mathsf{E}_{\mathsf{mal}} = \{\mathsf{dec}(\mathsf{enc}(x,y),y) = x,\ \mathsf{mal}(\mathsf{enc}(x,y),z) = \mathsf{enc}(z,y)\}.$$

The mal function symbol allows one to arbitrarily change the plaintext of an encryption. Such a malleable encryption is not realistic. It is only used for illustrative purpose.

By orienting from left to right the equations, we obtain a convergent rewrite system. Thus, $\mathsf{E}_{\mathsf{mal}}$ is a convergent equational theory. Let $\varphi = \{\mathsf{w}_1 \rhd \mathsf{enc}(s,\mathsf{k})\}$ where s and k are private constants. The only rule that is applicable is an instance of an **A** rule. Consider the rewrite rule $\mathsf{mal}(\mathsf{enc}(x,y),z) \to \mathsf{enc}(z,y)$ and the only deduction fact in $\mathrm{Init}(\varphi) = (\varphi, \emptyset)$. We obtain $r\sigma{\downarrow_{\mathcal{R}}} = \mathsf{enc}(z,\mathsf{k})$. This term is not ground and the condition required in case (1) is not fulfilled. Thus, we have that $\mathrm{Init}(\varphi) \Longrightarrow \bot$. Note that, since no other rule is applicable, there is no hope to find a strategy of rule applications to handle this case.

In this section, we identify a class of theories, called *layered convergent* theories, (a syntactically defined class of theories) for which failure is guaranteed not to occur.

6.1 Layered convergent theories

We prove that the algorithm never fails for *layered convergent* theories. Layered convergent theories consist in a generalization of subterm theories, considering each decomposition of the rewrite rules of the theory.

Definition 6.2 *(layered rewrite system).* A rewrite system $\mathcal{R}$, and by extension its equational theory E, are *layered* if there exists an ascending chain of subsets $\emptyset = \mathcal{R}_0 \subseteq \mathcal{R}_1 \subseteq \ldots \subseteq \mathcal{R}_{N+1} = \mathcal{R}$ $(N \geq 0)$, such that for every $0 \leq i \leq N$, for every rule $l \to r$ in $\mathcal{R}_{i+1} - \mathcal{R}_i$, for every (n,p,q)-decomposition $l = D[l_1,\ldots,l_n,y_1,\ldots,y_p,z_1,\ldots,z_q]$, one of the following two conditions holds:

(i) $\mathrm{var}(r) \subseteq \mathrm{var}(l_1,\ldots,l_n)$;
(ii) there exist $C_0, C_1, \ldots, C_k$ and $s_1,\ldots,s_k$ such that
—$r = C_0[s_1,\ldots,s_k]$;
—for each $1 \leq i \leq k$, $C_i[l_1,\ldots,l_n,y_1,\ldots,y_p,z_1,\ldots,z_q]$ rewrites to s_i in zero or one step of rewrite rule in head position along $\mathcal{R}_i$.

In the latter case, we say that the context $C = C_0[C_1,\ldots,C_k]$ is *associated* to the decomposition D of $l \to r$. Note that $C[l_1,\ldots,l_n,y_1,\ldots,y_p,z_1,\ldots,z_q] \to^*_{\mathcal{R}_i} r$.

The purpose of Condition (ii) is to ensure that a term $r\sigma$ coming from an instance of $l \to r \in \mathcal{R}_{i+1}$ is already deducible, applying rules of stricly smaller layers. More precisely, when a decomposition $l = D[l_1,\ldots,l_n,y_1,\ldots,y_p,z_1,\ldots,z_q]$ can be applied, then the resulting term r must be deducible from $l_1,\ldots,l_n,y_1,\ldots,y_p,z_1,\ldots,z_q$, possibly applying some (non nested) rewriting rules of $\mathcal{R}_i$ (of smaller layers).

The large class of weakly subterm convergent theories is an (easy) particular case of layered convergent theories.

LEMMA 6.3. *Any weakly subterm convergent rewrite system $\mathcal{R}$ is layered convergent.*

PROOF. Let $N = 0$ and $\mathcal{R}_1 = \mathcal{R}$. For any $l \to r$ in $\mathcal{R}$ and for every decomposition $l = D[l_1, \ldots, l_n, y_1, \ldots, y_p, z_1, \ldots, z_q]$, the term r is a subterm of l, thus either $r = C[l_1, \ldots, l_n, y_1, \ldots, y_p, z_1, \ldots, z_q]$ for some context C, or r is a subterm of some l_i thus $\mathrm{var}(r) \subseteq \mathrm{var}(l_1, \ldots, l_n)$. □

Consider the convergent theories of blind signatures $\mathsf{E}_{\mathsf{blind}}$ and prefix encryption $\mathsf{E}_{\mathsf{pref}}$ defined by the following sets of equations.

$$\mathcal{E}_{\mathsf{blind}} = \left\{ \begin{array}{rcl} \mathsf{checksign}(\mathsf{sign}(x,y),\mathsf{pub}(y)) & = & \mathsf{ok} \\ \mathsf{unblind}(\mathsf{blind}(x,y),y) & = & x \\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x,y),z),y) & = & \mathsf{sign}(x,z) \end{array} \right\}$$

$$\mathcal{E}_{\mathsf{pref}} = \mathcal{E}_{\mathsf{enc}} \cup \left\{ \mathsf{pref}(\mathsf{enc}(\langle x,y\rangle,z)) = \mathsf{enc}(x,z) \right\}$$

The theory $\mathsf{E}_{\mathsf{blind}}$ models primitives used in e-voting protocols [Delaune et al. 2009]. The prefix theory represents the property of many chained modes of encryption (e.g. CBC) where an attacker can retrieve any encrypted prefix out of a ciphertext.

LEMMA 6.4. *The rewrite system associated to the theory of homomorphism $\mathsf{E}_{\mathsf{hom}}$ defined in Section 2.3 as well as the rewrite systems obtained by orienting from left to right the equations in $\mathsf{E}_{\mathsf{blind}}$ and $\mathsf{E}_{\mathsf{pref}}$ are layered convergent.*

PROOF. Let us check for instance that the prefix theory $\mathsf{E}_{\mathsf{pref}}$ is layered. Let $N = 1$, $\mathcal{R}_1$ be the rewrite system obtained from $\mathcal{E}_{\mathsf{enc}}$ by orienting the equations from left to right, and $\mathcal{R}_2 = \mathcal{R}_1 \cup \{\mathsf{pref}(\mathsf{enc}(\langle x,y\rangle,z)) \to \mathsf{enc}(x,z)\}$. The rewrite rules of $\mathcal{R}_1$ satisfy the assumptions since $\mathcal{R}_1$ forms a convergent subterm rewrite system. The additional rule $\mathsf{pref}(\mathsf{enc}(\langle x,y\rangle,z)) \to \mathsf{enc}(x,z)$ admits three decompositions up to permutation of parameters:

—$l = \mathsf{pref}(l_1)$ and $D = \mathsf{pref}(\mathsf{w}_1)$, in which case $\mathrm{var}(r) \subseteq \mathrm{var}(l_1)$;
—$l = \mathsf{pref}(\mathsf{enc}(l_1, z))$ and $D = \mathsf{pref}(\mathsf{enc}(\mathsf{w}_1, \mathsf{w}_2))$. We have that $r = \mathsf{enc}(s_1, s_2)$ with $s_1 = x$ and $s_2 = z$. Consider the contexts $C_1 = \mathsf{proj}_1(\mathsf{w}_1)$ and $C_2 = \mathsf{w}_2$. We have that $C_1[l_1, z] \to_{\mathcal{R}_1} s_1$ and $C_2[l_1, z] = s_2$.
—$l = \mathsf{pref}(\mathsf{enc}(\langle x,y\rangle,z))$ and $D = \mathsf{pref}(\mathsf{enc}(\langle \mathsf{w}_1, \mathsf{w}_2\rangle), \mathsf{w}_3)$. We have that $r = \mathsf{enc}(s_1, s_2)$ with $s_1 = x$ and $s_2 = z$. Consider the contexts $C_1 = \mathsf{w}_1$ and $C_2 = \mathsf{w}_3$. We have that $C_1[x, y, z] = s_1$ and $C_2[x, y, z] = s_2$.

Verifying that the convergent theories $\mathsf{E}_{\mathsf{hom}}$ and $\mathsf{E}_{\mathsf{blind}}$ are layered is similar. □

6.2 A syntactic criterion

Definition 6.5 *(Maximal).* We say that the function Ctx is *maximal* if for every ϕ and t, if there exists s such that $\phi \vdash s$ and $t \to^*_{\mathcal{R}} s$, then $\mathrm{Ctx}(\phi \vdash^?_{\mathcal{R}} t) \neq \bot$.

We now prove that our algorithm never fails for *layered convergent* theories provided that the function Ctx in used is maximal. More precisely, we show that there exists no state (Φ, Ψ) from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation.

PROPOSITION 6.6. *Assume that the function* Ctx *in use is* maximal. *Then, provided that $\mathcal{R}$ is layered convergent, there exists no state (Φ, Ψ) from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation.*

PROOF. *Intuition.* We show this result by contradiction. Let (Φ, Ψ) be a state from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation, and let $l \to r$ be the rewrite rule involved in the corresponding instance of **A.3**. We prove the property by induction on the index $i \in \{0 \dots N\}$ such that $l \to r \in \mathcal{R}_{i+1} - \mathcal{R}_i$.

Actually, we show below that either Condition (i) stated in Definition 6.2 is satisfied and $r\sigma{\downarrow_{\mathcal{R}}}$ will be necessary ground, and thus **A.2** is applicable. Otherwise, the underlying decomposition satisfies Condition (ii) then, assuming that we have already treated the rules in $\mathcal{R}_i$ and relying on the fact that the function Ctx is maximal, we will deduce that **A.1** is applicable. In any case, we can avoid the application of **A.3**.

Full proof. More formally, using the notations of Figure 1 for the instance of **A.3** under consideration and the assumption on Ctx, we have that:

(a) for every $r\sigma \to_{\mathcal{R}}^* s$, $\Phi \cup \{z_1 \rhd z_1, \dots, z_q \rhd z_q\} \not\vdash s$, and
(b) $(r\sigma){\downarrow_{\mathcal{R}}}$ is not ground.

In particular, (b) implies that $\mathrm{var}(r)$ is not included in $\mathrm{var}(l_1, \dots, l_n)$, otherwise we would have

$$\begin{aligned} \mathrm{var}((r\sigma){\downarrow_{\mathcal{R}}}) &\subseteq \mathrm{var}(r\sigma) \subseteq \mathrm{var}(\mathrm{var}(r)\sigma) \\ &\subseteq \mathrm{var}(\mathrm{var}(l_1, \dots, l_n)\sigma) \subseteq \mathrm{var}(t_1, \dots, t_n) = \emptyset \end{aligned}$$

By assumption on the decomposition $l = D[l_1, \dots, l_n, y_1, \dots, y_p, z_1, \dots, z_q]$ of $l \to r \in \mathcal{R}_{i+1} - \mathcal{R}_i$, we deduce that there exist some contexts $C_0, \dots, C_k$ and some terms $s_1, \dots, s_k$ such that:

—$r = C_0[s_1, \dots, s_k]$;
—for each $1 \leq i \leq k$, $C_i[l_1, \dots, l_n, y_1, \dots, y_p, z_1, \dots, z_q]$ rewrites to s_i in zero or one step of rewrite rule in head position along $\mathcal{R}_i$.

Let $C = C_0[C_1, \dots, C_k]$ and $t_0 = C[l_1, \dots, l_n, y_1, \dots, y_p, z_1, \dots, z_q]$. Note that $t_0 \to_{\mathcal{R}_i}^* r$. If $t_0 = r$, we obtain that $r\sigma = C[t_1, \dots, t_{n+p}, z_1, \dots, z_q]$ is syntactically deducible from $\Phi \cup \{z_1 \rhd z_1, \dots, z_q \rhd z_q\}$, which contradicts (a). Hence $t_0 \to_{\mathcal{R}_i}^+ r$, and in particular $i > 0$.

Let μ be a substitution mapping the variables z_j to distinct fresh public constants a_j. For each $1 \leq i \leq k$, let $u_i = C_i[l_1, \dots, l_n, y_1, \dots, y_p, z_1, \dots, z_q]\sigma\mu$. The term $u_i = C_i[t_1, \dots, t_{n+p}, \mathsf{a}_1, \dots, \mathsf{a}_q]$ is syntactically deducible from Φ, and reduces to $u_i' = s_i\sigma\mu$ in zero or one step (in head position) along $\mathcal{R}_i$.

Case $u_i \to_{\mathcal{R}_i} u_i'$. By induction hypothesis on $i - 1$, no applicable rule **A.3** from (Φ, Ψ) may involve a rule in $\mathcal{R}_i$. Besides, by assumption, (Φ, Ψ) is saturated for the rules **B.1**, **B.2**, **A.1** and **A.2**. Therefore, Lemma 5.4 applied to $\Phi \vdash u_i$ and $u_i \to_{\mathcal{R}_i} u_i'$ implies that there exists u_i'' such that $u_i' \to_{\mathcal{R}}^* u_i''$ and $\Phi \vdash u_i''$.
Case $u_i' = u_i$. In such a case, trivially we have that there exists u_i'' such that $u_i' \to_{\mathcal{R}}^* u_i''$ and $\Phi \vdash u_i''$. We just have to choose $u_i'' = u_i'$.

Hence, in both cases, we have that there exists u_i'' such that $u_i' \to_{\mathcal{R}}^* u_i''$ and $\Phi \vdash u_i''$. Let $s = C_0[u_1'', \dots, u_k'']\mu^{-1}$ be the term obtained by replacing each a_i by z_i in $C[u_1'', \dots, u_k'']$. Since the a_i do not occur in $\mathcal{R}$ nor in Φ, we deduce that s satisfies

$r\sigma = C_0[s_1\sigma, \ldots, s_k\sigma] = C_0[u'_1, \ldots, u'_k]\mu^{-1} \to^*_{\mathcal{R}} s$ and $\Phi \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\} \vdash s$, in contradiction with the condition (a) stated at the beginning of the proof. □

6.3 Practical considerations.

Unfortunately, such a maximal Ctx is too inefficient in practice as one has to consider the syntactic deducibility problem $\phi \vdash s$ for every $t \to^*_{\mathcal{R}} s$. Proposition 6.7 below shows that the simple function context is actually sufficient to ensure non-failure when we know that another function Ctx already prevents failure on any state (reachable or not).

PROPOSITION 6.7. *Let $\mathcal{R}$ be a convergent rewrite system and* Ctx_0 *be an arbitrary function* Ctx*. If there exists no state (Φ, Ψ) from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation when the function* Ctx *in use is* Ctx_0*, then there exists no state (Φ, Ψ) from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation for any choice of* Ctx*.*

PROOF. Let Ctx_0 and Ctx'_0 be two arbitrary functions Ctx (*i.e.* they satisfy properties (a) and (b) defined on page 11). Assume that there exists no state (Φ, Ψ) from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation when the function Ctx in use is Ctx_0. Assume by contradiction that there exists a state (Φ_0, Ψ_0) from which $(\Phi_0, \Psi_0) \Longrightarrow \bot$ is the only applicable derivation for Ctx'_0. This means that there exist:

—a rewrite rule $l \to r \in \mathcal{R}$,
—a proper decomposition $D[l_1, \ldots, l_n, y_1, \ldots, y_p, z_1, \ldots, z_q]$ of l,
—some deduction facts $M_1 \rhd t_1, \ldots, M_{n+p} \rhd t_{n+p} \in \Phi_0$, and
—a substitution σ such that $(l_1, \ldots, l_n, y_1, \ldots, y_p)\sigma = (t_1, \ldots, t_{n+p})$.

Moreover, since this instance corresponds to an instance of **A.3**, we have that $r\sigma{\downarrow_{\mathcal{R}}}$ is not ground. When the function Ctx in use is Ctx_0, this instance has to correspond to an instance of **A.1** (**A.2** and **A.3** are impossible). Hence, we have that $\mathrm{Ctx}_0(\Phi_0 \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\} \vdash^?_{\mathcal{R}} r\sigma) \neq \bot$. This means that there exists s such that $r\sigma \to^*_{\mathcal{R}} s$ and $\Phi_0 \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\} \vdash s$. Since $\mathcal{R}$ is convergent, we have that $s \to^*_{\mathcal{R}} r\sigma{\downarrow_{\mathcal{R}}}$.

Let μ be a substitution mapping the variables z_j to distinct fresh public constants a_j. We have that $s\mu \to^*_{\mathcal{R}} (r\sigma{\downarrow_{\mathcal{R}}})\mu$ and also that $\Phi_0 \vdash s\mu$. Since $(\Phi_0, \Psi_0) \Longrightarrow \bot$ is the only applicable derivation for Ctx'_0, the rules **A.2**, **B.1**, and **B.2** cannot be applicable, even for Ctx_0. We saturate (Φ_0, Ψ_0) with the **A.1** rule for Ctx_0, reaching a state of the form (Φ_0, Ψ'_0) since only equations can be added to the state. Note also that the **A.1** rule can only be applied a finite a number of times and does not trigger the other rules. Thus (Φ_0, Ψ'_0) is saturated for Ctx_0. Using Lemma 5.4 (with the function Ctx_0), we obtain that $\Phi_0 \rhd (r\sigma{\downarrow_{\mathcal{R}}})\mu$, and thus $\Phi_0 \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\} \vdash r\sigma{\downarrow_{\mathcal{R}}}$. This contradicts the fact that **A.1** does not apply on (Φ_0, Ψ_0) when the function Ctx in use is Ctx'_0. Hence, the result. □

COROLLARY 6.8. *Let $\mathcal{R}$ be a layered convergent rewrite system and consider an arbitrary function* Ctx *in use. There exists no state (Φ, Ψ) from which $(\Phi, \Psi) \Longrightarrow \bot$ is the only applicable derivation.*

7. TERMINATION

In the previous section, we have described a sufficient criterion for non-failure. As shown by the example given below, this criterion does not ensure the termination of our saturation procedure.

Example 7.1. Consider the following layered convergent rewrite system $\mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{h}(x))$ where f is a public function symbol whereas g and h are private function symbols. Let $\varphi = \{\mathsf{w}_0 \rhd \mathsf{g}(\mathsf{a})\}$ where a is a private constant. By repeatedly applying the **A** rule on the newly generated deduction fact, we generate an infinite number of deduction facts of the form:

$$\mathsf{f}(\mathsf{w}_0) \rhd \mathsf{g}(\mathsf{h}(a)),\ \mathsf{f}(\mathsf{f}(\mathsf{w}_0)) \rhd \mathsf{g}(\mathsf{h}(\mathsf{h}(a)),\ \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{w}_0))) \rhd \mathsf{g}(\mathsf{h}(\mathsf{h}(\mathsf{h}(a))),\ \ldots$$

To obtain decidability for a given layered convergent theory, there remains only to provide a termination argument. Such an argument is generally easy to develop by hand as we illustrate on the example of the prefix theory. For the case of existing decidability results from [Abadi and Cortier 2006], such as the theories of blind signature and homomorphic encryption, we also provide a semantic criterion that allows us to directly conclude termination of the procedure. Note that this semantic criterion does not apply only to layered convergent theories but to any convergent theories (for which failure is guaranteed not to happen).

7.1 Termination of **B** rules

To begin with, we note that **B** rules always terminate after a polynomial number of steps. Let us write $\stackrel{\bullet}{\Longrightarrow}^n$ for the relation made of exactly n *strict applications* of rules ($S \stackrel{\bullet}{\Longrightarrow} S'$ iff $S \Longrightarrow S'$ and $S \neq S'$).

PROPOSITION 7.2. *For every states $S = (\Phi, \Psi)$ and S' such that $S \stackrel{\bullet}{\Longrightarrow}^n S'$ using only **B** rules, n is polynomially bounded in the size of* $\mathrm{im}(\Phi)$.

This is due to the fact that frames are one-to-one and that the rule **B.2** only adds deduction facts $M \rhd t$ such that t is a subterm of an existing term in Φ.

7.2 Proving termination by hand

For proving termination, we observe that it is sufficient to provide a function s mapping each frame Φ to a finite set of terms $s(\Phi)$ including the subterms of $\mathrm{im}(\Phi)$ and such that rule **A.2** only adds deduction facts $M \rhd t$ satisfying $t \in s(\Phi)$.

For subterm theories, we obtain polynomial termination by choosing $s(\Phi)$ to be the subterms of $\mathrm{im}(\Phi)$ together with the ground right-hand sides of $\mathcal{R}$.

PROPOSITION 7.3. *Let* E *be a weakly subterm convergent theory. For every $S = (\Phi, \Psi)$ and S' such that $S \stackrel{\bullet}{\Longrightarrow}^n S'$, n is polynomially bounded in the size of* $\mathrm{im}(\Phi)$.

To conclude that deduction and static equivalence are decidable in polynomial time [Abadi and Cortier 2006], we need to show that the deduction facts and the equations are of polynomial size. This requires a DAG representation for terms and visible equations. For our implementation, we have chosen not to use DAGs for the sake of simplicity since DAGs require much heavier data structures. However,

similar techniques as those described in [Abadi and Cortier 2006] would apply to implement our procedure using DAGs.

For proving termination for the prefix theory $\mathsf{E}_{\mathsf{pref}}$, it suffices to consider $s(\Phi) = \mathrm{st}_{\mathsf{pref}}(\Phi)$, where the notion $\mathrm{st}_{\mathsf{pref}}$ is recursively defined as follows:

—$\mathrm{st}_{\mathsf{pref}}(a) = \{a\}$ if a is a constant
—$\mathrm{st}_{\mathsf{pref}}(f(t_1,\ldots,t_n)) = \{f(t_1,\ldots,t_n)\} \cup \bigcup_{i=1}^{n} \mathrm{st}_{\mathsf{pref}}(t_i)$ $f \in \{\mathsf{dec}, \langle,\rangle, \mathsf{proj}_1, \mathsf{proj}_2, \mathsf{pref}\}$
—$\mathrm{st}_{\mathsf{pref}}(\mathsf{enc}(t,u)) = \{\mathsf{enc}(t,u)\} \cup \mathrm{st}_{\mathsf{pref}}(t) \cup \mathrm{st}_{\mathsf{pref}}(\mathsf{enc}(t_1,u))$ if $t = \langle t_1, t_2\rangle$
—$\mathrm{st}_{\mathsf{pref}}(\mathsf{enc}(t,u)) = \{\mathsf{enc}(t,u)\} \cup \mathrm{st}_{\mathsf{pref}}(t) \cup \mathrm{st}_{\mathsf{pref}}(u)$ otherwise.

PROPOSITION 7.4. *Consider the prefix theory* $\mathsf{E}_{\mathsf{pref}}$. *For every* $S = (\Phi, \Psi)$ *and* S' *such that* $S \stackrel{\cdot}{\Longrightarrow}^{n} S'$, n *is polynomially bounded in the size of* $\mathrm{im}(\Phi)$.

We then deduce that deduction and static equivalence are decidable for the equational theory $\mathsf{E}_{\mathsf{pref}}$. For this theory, deduction was already known to be decidable (in polynomial time) [Chevalier et al. 2003b]. However, decidability of static equivalence was not known before.

COROLLARY 7.5. *Deduction and static equivalence are decidable in polynomial time for the equational theory* $\mathsf{E}_{\mathsf{pref}}$.

Similarly, we may retrieve decidability of deduction and static equivalence for $\mathsf{E}_{\mathsf{hom}}$, $\mathsf{E}_{\mathsf{blind}}$, and $\mathsf{E}_{\mathsf{add}}$ defined in [Abadi and Cortier 2006]. For the theories $\mathsf{E}_{\mathsf{hom}}$ and $\mathsf{E}_{\mathsf{add}}$, it is actually sufficient to consider the notion of syntactic subterms. In order to conclude for the theory $\mathsf{E}_{\mathsf{blind}}$, we consider $\mathrm{st}_{\mathsf{blind}}$ defined as follows:

—$\mathrm{st}_{\mathsf{blind}}(a) = \{a\}$ if a is a constant
—$\mathrm{st}_{\mathsf{blind}}(f(t_1,\ldots,t_n)) = \{f(t_1,\ldots,t_n)\} \cup \bigcup_{i=1}^{n} \mathrm{st}_{\mathsf{blind}}(t_i)$ $f \in \{\mathsf{blind}, \mathsf{unblind}, \mathsf{checksign}\}$
—$\mathrm{st}_{\mathsf{blind}}(\mathsf{sign}(t,u)) = \{\mathsf{sign}(t,u)\} \cup \mathrm{st}_{\mathsf{blind}}(t) \cup \mathrm{st}_{\mathsf{blind}}(\mathsf{sign}(t_1,u))$ if $t = \mathsf{blind}(t_1,t_2)$
—$\mathrm{st}_{\mathsf{blind}}(\mathsf{sign}(t,u)) = \{\mathsf{sign}(t,u)\} \cup \mathrm{st}_{\mathsf{blind}}(t) \cup \mathrm{st}_{\mathsf{blind}}(u)$ otherwise.

7.3 A semantic criterion

We now provide a semantic criterion that ensures termination. This criterion intuitively states that the set of deducible terms from any initial frame φ should be equivalent to a set of *syntactically* deducible terms. Provided that failures are prevented and assuming a *fair* strategy for rule application, we prove that this criterion is a necessary and sufficient condition for our procedure to terminate.

Definition 7.6 *(Fair derivation).* A (possibly infinite) derivation

$$(\Phi_0, \Psi_0) \Longrightarrow \ldots \Longrightarrow (\Phi_n, \Psi_n) \Longrightarrow \ldots$$

is *fair* iff along this derivation,

(a) **B** rules are applied with greatest priority, and
(b) whenever a **A.i** rule is applicable for some instance $(l \to r, D, t_1, \ldots, t_n, \ldots)$, eventually the same instance of a rule **A.j** (with possibly $i \neq j$) is applied during the derivation.

Note that, for condition (b), it might be the case that **A.3** is applicable for some instance and later on, it is finally **A.1** that is aplied for this instance. Fairness implies that any deducible term is eventually syntactically deducible. This result follows from Lemma 5.3 and Lemma 5.4.

LEMMA 7.7. *Let $S_0 = (\Phi_0, \Psi_0) \Longrightarrow \ldots \Longrightarrow (\Phi_n, \Psi_n) \Longrightarrow \ldots$ be an infinite fair derivation from a state S_0. For every ground term t such that $\Phi_0 \vdash_{\mathsf{E}} t$, either $(\Phi_0, \Psi_0) \Longrightarrow^* \bot$ or there exists i such that $\Phi_i \vdash t{\downarrow_{\mathcal{R}}}$.*

PROOF. *Intuition.* Let t be a ground term deducible from Φ_i modulo E. There exists t_0 such that $M \rhd_{\Phi_i} t_0$ and $t_0 \to^* t{\downarrow_{\mathcal{R}}}$. This means that there exist a (public) context C and some deduction facts $M_1 \rhd t_1, \ldots, M_n \rhd t_n \in \Phi_i$ such that $M = C[M_1, \ldots, M_n]$ and $t_0 = C[t_1, \ldots, t_n]$. If t_0 is in normal form then we are done. Otherwise, this means that $t_0 \to t' \to^* t{\downarrow_{\mathcal{R}}}$. In a fair derivation, we know that we will consider this rewriting step at some point along the derivation. Applying Lemma 5.4, we will obtain that there exists t'' such that $t_0 \to t' \to^* t''$. Then, we will conclude by applying our induction hypothesis on t''.

Full proof. More formally, we show that either $(\Phi_i, \Psi_i) \Longrightarrow^* \bot$ or there exists $j \geq i$ such that $t{\downarrow_{\mathcal{R}}}$ is syntactically deducible from Φ_j, by induction on t_0 equipped with the order $<$ induced by the rewriting relation (that is $t_1 < t_2$ if and only if $t_2 \to^+ t_1$).
Base case: $t_0 = t{\downarrow_{\mathcal{R}}}$. In such a case, since $\Phi_i \vdash t_0$, we have that $\Phi_i \vdash t{\downarrow_{\mathcal{R}}}$. This allows us to conclude.

Induction step: $t_0 \to t' \to^* t{\downarrow_{\mathcal{R}}}$.

Along a fair derivation, **B** rules are applied in priority. Hence, we choose the smallest $i_1 \geq i$ such that no more **B** rules can be applied from (Φ_{i_1}, Ψ_{i_1}). Note indeed that there is no infinite derivation with only **B** rules (Proposition 7.2). We have still that $C[M_1, \ldots, M_n] \rhd_{\Phi_{i_1}} t_0 \to t'$.

Applying Lemma 5.4 and observing that no **B** rule can be applied from (Φ_{i_1}, Ψ_{i_1}), we are in one of the following cases:

—$(\Phi_{i_1}, \Psi_{i_1}) \Longrightarrow \bot$. In such a case, we easily conclude since $(\Phi_0, \Psi_0) \Longrightarrow^* \bot$.
—$\Phi_{i_1} \vdash t''$ for some t'' such that $t' \to^*_{\mathcal{R}} t''$. In such a case, we conclude by applying our induction hypothesis since $t'' < t' < t_0$. There exists $j \geq i_1$ such that $\Phi_j \vdash t{\downarrow_{\mathcal{R}}}$.
—Otherwise an instance $(l \to r,\ D,\ t_1, \ldots, t_n,\ \ldots)$ of a **A** rule is applicable. Note that this instance is entirely determined by the rewrite rule $l \to r$ involved in the rewriting step $t_0 \to t'$, the deduction facts $M_i \rhd t_i$ $(1 \leq i \leq n)$ and the public context that witness the fact that $\Phi_i \vdash t_0$.

By fairness, we know that a **A** rule will be applied along the derivation for the same instance $(l \to r,\ D,\ t_1, \ldots, t_n,\ \ldots)$. Let i_2 be the index on which this instance is applied. We have that $i_2 \geq i_1$. Note that since **B** rules are applied in priority, (Φ_{i_2}, Ψ_{i_2}) is saturated for **B** rules. Either, we have that $(\Phi_{i_2}, \Psi_{i_2}) \Longrightarrow \bot$ (and thus $(\Phi_i, \Psi_i) \Longrightarrow^* \bot$) or $(\Phi_{i_2}, \Psi_{i_2}) \Longrightarrow (\Phi_{i_2+1}, \Psi_{i_2+1})$.

We have that $C[M_1, \ldots, M_n] \rhd_{\Phi_{i_2}} t_0$ and $t_0 \to_{\mathcal{R}} t'$. By Lemma 5.4 and observing that no **B** rule can be applied from (Φ_{i_2}, Ψ_{i_2}), either $(\Phi_{i_2}, \Psi_{i_2}) \Longrightarrow \bot$ or there exists $(\Phi'_{i_2}, \Psi'_{i_2})$, M' and t'' such that:

—$(\Phi_{i_2}, \Psi_{i_2}) \Longrightarrow (\Phi'_{i_2}, \Psi'_{i_2})$;
—$M' \rhd_{\Phi'_{i_2}} t''$ with $t' \to^*_{\mathcal{R}} t''$; and
—$\Psi'_{i_2} \models C[M_1, \ldots, M_n] \bowtie M'$.

Actually, the instance of the **A** rule that is applied in this derivation is entirely determined by the rewrite rule $l \to r$ involved in the rewriting step $t_0 \to t'$, the public context C and the deduction facts $M_i \rhd t_i$ $(1 \leq i \leq n)$ that witness the fact that $\Phi_i \vdash t_0$ (and thus $\Phi_{i_2} \vdash t_0$). Hence, we have that $(\Phi'_{i_2}, \Psi'_{i_2}) = (\Phi_{i_2+1}, \Psi_{i_2+1})$.

Thus we have that $M' \rhd_{\Phi'_{i_2+1}} t''$ with $t'' \to^* t{\downarrow_{\mathcal{R}}}$ and $t'' < t' < t$. We can apply our induction hypothesis, either $(\Phi_{i_2+1}, \Psi_{i_2+1}) \Longrightarrow^* \bot$ (and thus $(\Phi_i, \Psi_i) \Longrightarrow^* \bot$) or there exists $j \geq i_2 + 1$ such that $\Phi_j \vdash t{\downarrow_{\mathcal{R}}}$. □

Our termination criterion (Property (*ii*) below) is a semantic criterion. It is related to the notion *locally stable* introduced in [Abadi and Cortier 2006].

Proposition 7.8 criterion for termination. *Let φ be an initial frame such that* $\mathrm{Init}(\varphi) \not\Longrightarrow^* \bot$. *The following conditions are equivalent:*

(i) There exists a saturated pair (Φ, Ψ) *such that* $\mathrm{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$.
(ii) There exists a (finite) initial frame φ_s such that for every term t, t is deducible from φ modulo E *iff $t{\downarrow_{\mathcal{R}}}$ is syntactically deducible from φ_s.*
(iii) There exists no fair infinite derivation starting from $\mathrm{Init}(\varphi)$.

Proof. (*iii*) ⇒ (*i*): trivial. Indeed by using a fair derivation we will eventually reach a weakly saturated state. (*i*) ⇒ (*ii*): Let $\Phi = \{M_1 \rhd s_1, \ldots, M_\ell \rhd s_\ell\}$ and $\varphi_s = \{\mathsf{w}_1 \rhd s_1, \ldots, \mathsf{w}_\ell \rhd s_\ell\}$. Let t be a ground term. By Theorem 4.6, we have that $\exists M \,.\, M \rhd^{\mathsf{E}}_{\varphi} t$ iff $\exists M \,.\, M \rhd_{\Phi} t{\downarrow_{\mathcal{R}}}$, i.e. $\exists M \,.\, M \rhd_{\varphi_s} t{\downarrow_{\mathcal{R}}}$. (*ii*) ⇒ (*iii*): we need to prove that there exists no fair infinite derivation starting from $\mathrm{Init}(\varphi)$.

Let $\varphi_s = \{\mathsf{w}_1 \rhd s_1, \ldots, \mathsf{w}_\ell \rhd s_\ell\}$ an initial frame such that for every t, $\exists M \,.\, M \rhd^{\mathsf{E}}_{\varphi} t$ is equivalent to $\exists M \,.\, M \rhd_{\varphi_s} t{\downarrow_{\mathcal{R}}}$. Assume by contradiction that there is an infinite fair derivation $(\Phi_0, \Psi_0) \Longrightarrow \ldots \Longrightarrow (\Phi_n, \Psi_n) \Longrightarrow \ldots$ with $(\Phi_0, \Psi_0) = \mathrm{Init}(\varphi)$.

By Lemma 7.7 and since $\mathrm{Init}(\varphi) \not\Longrightarrow^* \bot$, we deduce that there exists i_0 such that each s_i, $1 \leq i \leq \ell$ is syntactically deducible from Φ_{i_0}. Since there is no infinite derivation with only **B** rules (Proposition 7.2), we can also assume that no **B** rule can be applied from Φ_{i_0}. We have that $\exists M \,.\, M \rhd^{\mathsf{E}}_{\varphi} t$ is now equivalent to $\exists M \,.\, M \rhd_{\Phi_{i_0}} t{\downarrow_{\mathcal{R}}}$. Thanks to the property (a) of the function Ctx, we know that the function Ctx will not return $\bot$. This implies that the **A.2** rule cannot be applied either. We deduce that no deduction facts are added to Φ_{i_0} along the derivation, that is $\Phi_j = \Phi_{i_0}$ for every $j \geq i_0$. Since no deduction fact are added, only a finite number of **A.1** rules can be applied, which contradicts the existence of an infinite chain. □

Assuming a theory for which our algorithm does not fail, this criterion (Property (*ii*)) shows that termination is equivalent to the local stability criterion defined in [Abadi and Cortier 2006]. Note however that this semantic criterion cannot be used together with the syntactic criterion described in Section 6 to establish decidability of deduction and static equivalence for layered convergent theories even if they belong to the class of locally stable theories defined in [Abadi and Cortier

2006]. Indeed, Proposition 7.8 requires that our algorithm never fail whereas Corollary 6.8 only states that, for layered convergent theories, it is not possible to reach a state from which failure is the only possible option.

8. IMPLEMENTATION: THE TOOL YAPA

YAPA (Yet Another Protocol Analyzer) is an Ocaml implementation of the saturation procedure presented in Section 4 with several optional optimizations. It can be freely downloaded[2] together with a brief manual and examples.

The tool takes as input an equational theory described by a finite convergent rewrite system, as well as frame definitions and queries. The procedure starts by computing the decompositions of the rewrite system. By default, the following optimization is done: provided that the rewrite rules are given in an order compatible with the sets $\mathcal{R}_0 \subseteq \ldots \subseteq \mathcal{R}_{N+1}$ of Definition 6.2, the tool is able to recognize layered theories and to pre-compute the associated contexts C related to condition (ii) of this definition. This allows resolving the failure cases as soon as they appear, rather than later on, when the saturation procedure has made enough progress. This optimization was studied in a first version of this article [Baudet et al. 2009] but as the practical benefits appear to be minor (see below), we chose not to keep these technical developments in this version for the sake of notational simplicity.

Another optimization concerns a specific treatment of subterm convergent theories but does not induce any difference with the theoretical procedure presented here. Except for the first (optional) optimization mentioned above, the algorithm follows the procedure described in Section 4, using a minimal function Ctx in the sense in Section 6.3, and a fair strategy of rule application (see Definition 7.6).

We have conducted several experiments on a PC Intel Core 2 Duo at 2.4 GHz with 2 Go RAM for various equational theories (see below) and found that YAPA provides an efficient way to check static equivalence and deducibility. Those examples are available at `http://www.lsv.ens-cachan.fr/~baudet/yapa/index.html`. The figures given below are valid for the versions with and without optimizations.

For the case of $\mathsf{E}_{\mathsf{enc}}$, $\mathsf{E}_{\mathsf{hom}}$, and $\mathsf{E}_{\mathsf{pref}}$, we have run YAPA on the frames:

—$\varphi_n = \{\mathsf{w}_1 \rhd t_n^0, \mathsf{w}_2 \rhd \mathsf{c}_0, \mathsf{w}_3 \rhd \mathsf{c}_1\}$, and
—$\varphi'_n = \{\mathsf{w}_1 \rhd t_n^1, \mathsf{w}_2 \rhd \mathsf{c}_0, \mathsf{w}_3 \rhd \mathsf{c}_1\}$,

where $t_0^i = \mathsf{c}_i$ and $t_{n+1}^i = \langle \mathsf{enc}(t_n^i, \mathsf{k}_n^i), \mathsf{k}_n^i \rangle$, $i \in \{0, 1\}$. These examples allow us to increase the (tree, non-DAG) size of the distinguishing tests exponentially, while the sizes of the frames grow linearly. Despite the size of the output, we have observed satisfactory performances for the tool.

n	10	14	16	18	20
Execution time - $\mathsf{E}_{\mathsf{enc}}$	$<$ 1s	1,7s	8s	30s	$<$ 3min
Execution time - $\mathsf{E}_{\mathsf{pref}}$	$<$ 1s	2,3s	10s	43s	3min
Execution time - $\mathsf{E}_{\mathsf{hom}}$	$<$ 1 s	5,5s	26s	2min	10min

We have also experimented YAPA on the theory of addition $\mathsf{E}_{\mathsf{add}}$ defined in [Abadi and Cortier 2006] with the frames:

[2] `http://www.lsv.ens-cachan.fr/~baudet/yapa/index.html`

—$\varphi_n = \{\mathsf{w}_1 \rhd \mathsf{c}_0, \mathsf{w}_2 \rhd s^n(\mathsf{c}_0)\}$, and
—$\varphi'_n = \{\mathsf{w}_1 \rhd \mathsf{c}_0, \mathsf{w}_2 \rhd s^{n+1}(\mathsf{c}_0)\}$.

n	20	40	80	160
Execution time	$<$ 1s	$<$ 1s	2s	13s

Lastly, we have experimented YAPA on the theory $\mathsf{E}_{\mathsf{blind}}$ with the frames encountered in the study of privacy of the e-voting protocol by Fujioka, Okamoto, and Ohta [Fujioka et al. 1992]. Those frames are described in [Delaune et al. 2009] and YAPA always answers in less than 1 seconde, even for frames that describe the execution for up to 20 voters.

Comparison with ProVerif. In comparison with the tool ProVerif [Blanchet 2001; Blanchet et al. 2008], here instrumented to check static equivalences, our test samples suggest a running time between one and two orders of magnitude faster for YAPA. Also we did not succeed in making ProVerif terminate on the two theories $\mathsf{E}_{\mathsf{hom}}$ and $\mathsf{E}_{\mathsf{add}}$. Of course, these results are not entirely surprising given that ProVerif is tailored for the more general (and difficult) problem of protocol (in)security under active adversaries. In particular ProVerif's initial preprocessing of the rewrite system appears more substantial than ours and does not terminate on the theories $\mathsf{E}_{\mathsf{hom}}$ and $\mathsf{E}_{\mathsf{add}}$. However, ProVerif handles some theories that YAPA does not handle (e.g. the theory $\mathsf{E}_{\mathsf{mal}}$ given in Example 6.1) and some non-convergent theories such as commutativity and the equation $\mathsf{exp}(\mathsf{exp}(\mathsf{g}, x), y) = \mathsf{exp}(\mathsf{exp}(\mathsf{g}, y), x)$ which can be used as a basic model of Diffie-Hellman, so the set of equational theories supported by YAPA and ProVerif are incomparable.

Comparison with KiSs. The tool KiSs (Knowledge in Security protocolS) is a C++ implementation of the procedure described in [Ciobâcă et al. 2009]. This procedure reused the same concepts than the one presented in a preliminary version of this work [Baudet et al. 2009], relying however on a more general representation of deduction facts. Actually, our procedure represents the set of deducible terms by the means of a finite set of ground terms S. Deducible terms are those in S and those obtained by applying public function symbols on terms in S. When it is not possible to represent all the deducible terms with such a representation, our procedure fails. This is exactly what happens for the theory $\mathsf{E}_{\mathsf{mal}}$. Coming back to Example 6.1, the set of deducible terms from $\varphi = \{w_1 \rhd \mathsf{enc}(s, k)\}$ contains at least all the terms of the form $\mathsf{enc}(x\sigma, k)$ where $x\sigma$ reprensents an arbitrary deducible term. However, note that k itself is not deducible. Hence, our representation is not suitable to represent this set of deducible terms.

To overcome this limitation, one possibility is to consider a more involved representation for deducible terms. The procedure implemented in KiSs allows one to consider deduction facts with side conditions [Ciobâcă et al. 2009]. Moreover, the deduction facts used in KiSs include terms with variables that can be substituted by any deducible terms. This allows one to consider the fact $[\mathsf{mal}(\mathsf{w}_1, X) \rhd \mathsf{enc}(x, \mathsf{k}) \mid X \rhd x]$ that intuitively exactly represents the set described above. However, because of this quite general representation of deducible terms, it is more complicated to ensure termination of the procedure implemented in KiSs. Except for the class of subterm convergent equational theories, they do not provide any general syntactic criterion [Ciobâcă et al. 2009].

The performances of the tool YAPA are comparable to the performances of KiSs. However, since the tool KiSs implements DAG representations for terms, it does better on the example developed above. Moreover, KiSs allows one to consider some equational theories for which our procedure fails (e.g. the theory of trapdoor bit commitment).

9. CONCLUSION AND FUTURE WORK

We have proposed a procedure for checking deducibility and static equivalence. Our procedure is correct and complete for any convergent theory and is efficient, as shown by its implementation within the tool YAPA. Since deducibility and static equivalence are undecidable in general, our algorithm may fail or may not terminate. We have identified a large class of equational theories (called layered convergent) for which non-failure of the procedure is ensured. Since termination can then often be easily proved by hand, we have obtained a new decidability result for the prefix theory and retrieved decidability for the convergent theories defined in [Abadi and Cortier 2006]. We have also proposed a semantic (and exact) characterization for the procedure to terminate.

As further work, we would like to extend our procedure to theories with associative and commutative operators. A first possibility would be to implement the decidability result of [Cortier and Delaune 2007] for monoidal theories (that include many theories with associative and commutative operators) and to combine the two procedures using the combination theorem of [Arnaud et al. 2007]. However, it seems much more efficient to integrate associativity and commutativity directly and this could even open the way to a more powerful combination technique.

The tool KiSs, developed recently [Ciobâcă et al. 2009], supports several equational theories for which our procedure fails. Conversely our procedure is guaranteed to terminate (without failure) for the prefix theory while it is not known whether KiSs always terminates for this theory. More generally, proving termination in YAPA is usually easy and immediately yields decidability for layered convergent theories, while proving termination in KiSs is more involved. It would be interesting to compare the techniques and possibly to combine them in order to capture more theories.

Lastly, as indicated in the introduction, deduction and static equivalence are static notions. They do not take into account the dynamic behaviour of the underlying protocols. Even if these notions play an important role for the analysis of security protocols in presence of an active attacker, it remains challenging to obtain decidability results for the active case in presence of algebraic properties. There are already several tools for deciding various trace-based security properties (e.g. secrecy and authentication) in presence of algebraic properties. Regarding equivalence-based security properties, several procedures exist [Blanchet et al. 2008; Baudet 2005; Tiu and Dawson 2010; Cheval et al. 2010; Chevalier and Rusinowitch 2010]. However, most of them have not yet been implemented and/or do not deal with algebraic properties. Only the ProVerif tool [Blanchet et al. 2008] allows one to establish automatically equivalence-based properties in presence of algebraic properties considering however a very strong notion of equivalence.

Acknowledgments

We are very grateful to the anonymous reviewers for their careful reading and helpful suggestions

REFERENCES

Abadi, M., Baudet, M., and Warinschi, B. 2006. Guessing attacks and the computational soundness of static equivalence. In *Foundations of Software Science and Computation Structures (FOSSACS'06)*. 398–412.

Abadi, M. and Cortier, V. 2006. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science 387,* 1-2, 2–32.

Abadi, M. and Fournet, C. 2001. Mobile values, new names, and secure communication. In *28th ACM Symposium on Principles of Programming Languages (POPL'01)*. ACM, 104–115.

Anantharaman, S., Narendran, P., and Rusinowitch, M. 2007. Intruders with caps. In *18th Conference on Term Rewriting and Applications (RTA'07)*. LNCS, vol. 4533. Springer.

Arapinis, M., Chothia, T., Ritter, E., and Ryan, M. 2009. Untraceability in the applied pi calculus. In *Proceeding of the 1st International Workshop on RFID Security and Cryptography*.

Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J., Hankes Drielsma, P., Héam, P.-C., Kouchnarenko, O., Mantovani, J., Mödersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Viganò, L., and Vigneron, L. 2005. The AVISPA Tool for the automated validation of internet security protocols and applications. In *17th Conference on Computer Aided Verification, CAV'2005*. LNCS, vol. 3576. Springer, 281–285.

Arnaud, M., Cortier, V., and Delaune, S. 2007. Combining algorithms for deciding knowledge in security protocols. In *Proc. 6th International Symposium on Frontiers of Combining Systems (FroCoS'07)*. Lecture Notes in Artificial Intelligence, vol. 4720. Springer, 103–117.

Baudet, M. 2005. Deciding security of protocols against off-line guessing attacks. In *12th ACM Conference on Computer and Communications Security (CCS'05)*. ACM Press, 16–25.

Baudet, M. 2007. Thèse de doctorat. Ph.D. thesis, Laboratoire Spécification et Vérification, ENS Cachan, France.

Baudet, M., Cortier, V., and Delaune, S. 2009. YAPA: A generic tool for computing intruder knowledge. In *20th International Conference on Rewriting Techniques and Applications (RTA'09)*. Lecture Notes in Computer Science, vol. 5595. Springer, Brasília, Brazil, 148–163.

Baudet, M., Cortier, V., and Kremer, S. 2005. Computationally sound implementations of equational theories against passive adversaries. In *32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*. LNCS, vol. 3580. Springer, 652–663.

Blanchet, B. 2001. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *14th Computer Security Foundations Workshop (CSFW'01)*. IEEE Comp. Soc. Press, 82–96.

Blanchet, B., Abadi, M., and Fournet, C. 2008. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming 75,* 1, 3–51.

Boichut, Y., Héam, P.-C., and Kouchnarenko, O. 2006. Handling algebraic properties in automatic analysis of security protocols. In *Theoretical Aspects of Computing (ICTAC'06)*. Lecture Notes in Computer Science, vol. 4281. Springer, 153–167.

Cheval, V., Comon-Lundh, H., and Delaune, S. 2010. Automating security analysis: symbolic equivalence of constraint systems. In *5th International Joint Conference on Automated Reasoning (IJCAR'10)*. Lecture Notes in Artificial Intelligence, vol. 6173. Springer, 412–426.

Chevalier, Y., Küsters, R., Rusinowitch, M., and Turuani, M. 2003a. Deciding the security of protocols with Diffie-Hellman exponentiation and product in exponents. In *Proceedings of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*. LNCS, vol. 2914. Springer-Verlag, Mumbai (India), 124–135.

Chevalier, Y., Küsters, R., Rusinowitch, M., and Turuani, M. 2003b. An NP decision procedure for protocol insecurity with XOR. In *18th IEEE Symposium on Logic in Computer Science (LICS'03)*. IEEE Comp. Soc. Press.

Chevalier, Y. and Rusinowitch, M. 2010. Decidability of symbolic equivalence of derivations. *Journal of Automated Reasoning*.

Ciobâcă, Ş., Delaune, S., and Kremer, S. 2009. Computing knowledge in security protocols under convergent equational theories. In *22nd International Conference on Automated Deduction (CADE'09)*. Lecture Notes in Artificial Intelligence. Springer, 355–370.

Comon, H. and Shmatikov, V. 2002. Is it possible to decide whether a cryptographic protocol is secure or not? *Journal of Telecommunications and Information Technology 4/2002*, 5–15.

Comon-Lundh, H. and Cortier, V. 2003. New decidability results for fragments of first-order logic and application to cryptographic protocols. In *Proc. of the 14th Int. Conf. on Rewriting Techniques and Applications (RTA'2003)*. LNCS, vol. 2706. Springer-Verlag, 148–164.

Comon-Lundh, H. and Shmatikov, V. 2003a. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *18th IEEE Symposium on Logic in Computer Science (LICS'03)*. IEEE Comp. Soc. Press.

Comon-Lundh, H. and Shmatikov, V. 2003b. 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)*. IEEE Computer Society, Los Alamitos, CA, 271–280.

Corin, R., Doumen, J., and Etalle, S. 2004. Analysing password protocol security against off-line dictionary attacks. In *2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP'04)*. ENTCS.

Cortier, V. and Delaune, S. 2007. Deciding knowledge in security protocols for monoidal equational theories. In *14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07)*. LNAI. Springer.

Cortier, V., Delaune, S., and Lafourcade, P. 2006a. A Survey of Algebraic Properties Used in Cryptographic Protocols. *Journal of Computer Security 14,* 1/2006.

Cortier, V., Delaune, S., and Lafourcade, P. 2006b. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security 14,* 1, 1–43.

Cortier, V., Keighren, G., and Steel, G. 2007. Automatic analysis of the security of xor-based key management schemes. In *13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07)*. Lecture Notes in Computer Science, vol. 4424. Springer, Braga, Portugal, 538–552.

Cremers, C. J., Lafourcade, P., and Nadeau, P. 2009. Comparing state spaces in automatic protocol analysis. In *Formal to Practical Security*. Lecture Notes in Computer Science, vol. 5458/2009. Springer Berlin / Heidelberg, 70–94.

Delaune, S. and Jacquemard, F. 2004. A decision procedure for the verification of security protocols with explicit destructors. In *11th ACM Conference on Computer and Communications Security (CCS'04)*. 278–287.

Delaune, S., Kremer, S., and Ryan, M. D. 2009. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security 17,* 4 (July), 435–487.

Escobar, S., Meadows, C., and Meseguer, J. 2008. State space reduction in the maude-nrl protocol analyzer. In *13th European Symposium on Research in Computer Security (ESORICS08)*. Lecture Notes in Computer Science, vol. 5283. Springer, 548–562.

Fujioka, A., Okamoto, T., and Ohta, K. 1992. A practical secret voting scheme for large scale elections. In *Advances in Cryptology – AUSCRYPT '92*. LNCS, vol. 718. Springer, 244–251.

Goubault-Larrecq, J., Roger, M., and Verma, K. N. 2004. Abstraction and resolution modulo AC: How to verify Diffie-Hellman-like protocols automatically. *Journal of Logic and Algebraic Programming 64,* 2, 219–251.

Küsters, R. and Truderung, T. 2010. Reducing protocol analysis with XOR to the XOR-free case in the Horn theory based approach. *Journal of Automated Reasoning*. To appear.

Lowe, G. 1996. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*. LNCS, vol. 1055. Springer-Verlag, 147–166.

Millen, J. and Shmatikov, V. 2001. Constraint solving for bounded-process cryptographic protocol analysis. In *8th ACM Conference on Computer and Communications Security (CCS'01)*.

Mödersheim, S. and Viganò, L. 2009. The open-source fixed-point model checker for symbolic analysis of security protocols. In *Fosad 2007-2008-2009*. Lecture Notes in Computer Science. Springer, 166–194.

Seidl, H. and Verma, K. N. 2009. Flat and one-variable clauses for single blind copying protocols: The xor case. In *20th International Conference on Rewriting Techniques and Applications (RTA'09)*. Lecture Notes in Computer Science, vol. 5595. Springer, 118–132.

Shmatikov, V. 2004. Decidable analysis of cryptographic protocols with products and modular exponentiation. In *Proc. 13th European Symposium On Programming (ESOP'04)*. LNCS, vol. 2986. Springer-Verlag, Barcelona (Spain), 355–369.

Tiu, A. and Dawson, J. E. 2010. Automating open bisimulation checking for the spi calculus. In *23rd Computer Security Foundations Symposium (CSF'10)*. IEEE Computer Society, 307–321.

Turuani, M. 2006. The CL-Atse Protocol Analyser. In *Term Rewriting and Applications - Proc. of RTA*. Lecture Notes in Computer Science, vol. 4098. Seattle, WA, USA, 277–286.

Verma, K. N. 2003. Two-way equational tree automata for AC-like theories: Decidability and closure properties. In *Proc. 14th International Conference on Rewriting Techniques and Applications (RTA'03)*. LNCS, vol. 2706. Springer-Verlag, Valencia (Spain), 180–196.

A. APPENDIX

LEMMA 5.2 (COMPLETENESS, SYNTACTIC DEDUCTION). *Let (Φ, Ψ) be a state, $M_0 \rhd t_0 \in \Phi$. Let N, t be two terms such that $t \in \mathrm{st}(t_0)$ and $N \rhd_\Phi t$. Then there exists (Φ', Ψ') and N' such that:*

—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$ *using* $\boldsymbol{B}$ *rules, and*
—$N' \rhd t \in \Phi'$ *and* $\Psi' \models N \bowtie N'$.

PROOF. By hypothesis, we have that $N \rhd_\Phi t$. This means that there exists a public context C and some facts $M_1 \rhd t_1, \ldots, M_n \rhd t_n \in \Phi$ such that $N = C[M_1, \ldots, M_n]$ and $t = C[t_1, \ldots, t_n]$. Let C be such a context whose size is minimal. We show the result by structural induction on C.

Base case: C is reduced to an hole. Let $(\Phi', \Psi') = (\Phi, \Psi)$ and $N' = N$. The result trivially holds.

Induction step: $C = f(C_1, \ldots, C_r)$ with $f \in \mathcal{F}_{\mathsf{pub}}$ of arity r. In such a case, we have $t = f(u_1, \ldots, u_r)$ and $C_i[M_1, \ldots, M_n] \rhd_\Phi u_i$ with $u_i \in \mathrm{st}(t_0)$ for each $1 \leq i \leq r$. Thus, we can repeatedly apply our induction hypothesis. First for $i = 1$. Then you can apply again our induction hypothesis to extend the resulting derivation (with $i = 2$), and so on until $i = r$. At the end, we deduce that there exists (Φ_1, Ψ_1) and terms $N'_1, \ldots N'_r$ such that:

—$(\Phi, \Psi) \Longrightarrow^* (\Phi_1, \Psi_1)$ using $\mathbf{B}$ rules,
—$N'_i \rhd u_i \in \Phi_1$ and $\Psi_1 \models C_i[M_1, \ldots, M_n] \bowtie N'_i$ for each $1 \leq i \leq r$.

From this we easily deduce that $\Psi_1 \models N \bowtie f(N'_1, \ldots, N'_r)$. We apply one $\mathbf{B}$ rule. We have that $M_0 \rhd t_0, N'_1 \rhd u_1, \ldots, N'_r \rhd u_r \in \Phi_1$, $t = f(u_1, \ldots, u_r) \in \mathrm{st}(t_0)$ and $f \in \mathcal{F}_{\mathsf{pub}}$. We distinguish two cases:

Rule ***B.1***. Assume that there exists M_t such that $M_t \rhd t \in \Phi_1$.

Let $\Phi' = \Phi_1$, $\Psi' = \Psi_1 \cup \{f(N'_1, \ldots, N'_r) \bowtie M_t\}$ and $N' = M_t$. In order to conclude it remains to show that $\Psi' \models N \bowtie N'$. We have $\Psi' \models f(N'_1, \ldots, N'_r) \bowtie N'$ and $\Psi' \models N \bowtie f(N'_1, \ldots, N'_r)$. This allows us to conclude.

Rule ***B.2***: Assume that for all M_t we have that $(M_t \rhd t) \notin \Phi_1$.

Let $\Phi' = \Phi_1 \cup \{f(N'_1, \ldots, N'_r) \rhd t\}$, $\Psi' = \Psi_1$ and $N' = f(N'_1, \ldots, N'_r)$. In order to conclude it remains to show that $\Psi' \models N \bowtie N'$. This is an easy consequence of the fact that $\Psi_1 \models N \bowtie f(N'_1, \ldots, N'_r)$. □

LEMMA 5.3 (COMPLETENESS, SYNTACTIC EQUATIONS). *Let (Φ, Ψ) be a state, and M, N be two terms such that $M \rhd_\Phi t$ and $N \rhd_\Phi t$ for some term t. Then there exists (Φ', Ψ') such that:*

—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$ *using* $\boldsymbol{B}$ *rules, and*
—$\Psi' \models M \bowtie N$.

PROOF. By hypothesis, we have that $M \rhd_\Phi t$ and $N \rhd_\Phi t$ for some term t. By definition of $\rhd_\Phi$, we have that

—$M = C[M_1, \ldots, M_k]$, $N = C'[N_1, \ldots, N_\ell]$ for some contexts C, C',

—the facts $M_1 \rhd t_1, \ldots, M_k \rhd t_k$ and $N_1 \rhd u_1, \ldots, N_\ell \rhd u_\ell$ are in Φ,
—$C[t_1, \ldots, t_k] = C'[u_1, \ldots, u_\ell]$.

We prove the result by structural induction on C and C'. We assume w.l.o.g. that C is smaller than C' (in terms of number of symbols).

Base case: C is reduced to a hole. We have that $C[M_1, \ldots, M_k] = M_1$. By hypothesis, we have that $N \rhd_\Phi t = t_1$ and thus $t \in \mathrm{st}(t_1)$. Thanks to Lemma 5.2, there exists (Φ', Ψ') and N' such that $(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$ using a **B** rule, $N' \rhd t_1 \in \Phi'$ and $\Psi' \models N \bowtie N'$. Since $M_1 \rhd t_1$ and $N' \rhd t_1$ are both in Φ', we deduce that $N' = M_1$. Hence we have that $N' = M$ and thus we easily conclude.

Induction step: $C = f(C_1, \ldots, C_r)$ and $C' = f(C'_1, \ldots, C'_r)$ where $f \in \mathcal{F}_{\mathsf{pub}}$ is a symbol of arity r and $C_1, \ldots, C_r, C'_1, \ldots, C'_r$ are contexts. Moreover, we have that $C_i[t_1, \ldots, t_k] = C'_i[u_1, \ldots, u_\ell]$ for every $1 \leq i \leq r$. Thus, we can repeatedly apply our induction hypothesis. First for $i = 1$ resulting in a derivation that can then be extended by applying our induction hypothesis (with $i = 2$), and so on until $i = r$. At the end, we deduce that there exists (Φ', Ψ') such that:

—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$, and
—$\Psi' \models C_i[M_1, \ldots, M_k] \bowtie C'_i[N_1, \ldots, N_\ell]$ for every $1 \leq i \leq r$.

Hence, we have that $\Psi' \models M \bowtie N$. This allows us to conclude. □

The following lemma justifies the notion of decomposition (Definition 4.1) as far as completeness is concerned.

LEMMA A.1 DECOMPOSITION OF A CONTEXT REDUCTION. *Let Φ be a frame, l a (plain) term, σ a substitution, and M a term such that $M \rhd_\Phi l\sigma$. Then there exist*

—*a (n, p, q)-decomposition D of l, written $l = D[l_1, \ldots, l_n, y_1, \ldots y_{p+q}]$,*
—*n deduction facts $M_1 \rhd t_1, \ldots, M_n \rhd t_n$ in Φ,*
—*$p + q$ recipes $N_1, \ldots, N_{p+q}$*

such that

—*for every $1 \leq i \leq n$, $t_i = l_i\sigma$ and*
—*for every $1 \leq j \leq p + q$, $N_j \rhd_\Phi y_j\sigma$.*

In particular, $D[M_1, \ldots, M_n, N_1, \ldots N_{p+q}] \rhd_\Phi l\sigma$.

Besides, if l is a left-hand side of rule in $\mathcal{R}$ and Φ is $\mathcal{R}$-reduced, D is a proper decomposition (i.e. $D \neq \mathsf{w}_1$).

PROOF. Since $M \rhd_\Phi l\sigma$, by definition there exists C and $M^0_1 \rhd t^0_1, \ldots, M^0_m \rhd t^0_m$ in Φ such that $M = C[M^0_1, \ldots, M^0_m]$ and $l\sigma = C[t^0_1, \ldots, t^0_m]$.

Let $x_1, \ldots, x_m$ be fresh variables. Given that $C[x_1, \ldots, x_m]$ and l unify and have distinct variables, there exists a largest common context D_0 such that $l = D_0[l^0_1, \ldots, l^0_a, y^0_1, \ldots, y^0_b]$ and $C = D_0[\mathsf{w}_{j_1}, \ldots, \mathsf{w}_{j_a}, D_1, \ldots, D_b]$ where the terms l^0_i are not variables and D_0 uses all his parameters: in particular $l\sigma = C[t^0_1, \ldots, t^0_m]$ means that

—for every $1 \leq k \leq a$, $l^0_k\sigma = t^0_{j_k}$, and

—for every $1 \leq k \leq b$, $y^0_k\sigma = D_k[t^0_1, \ldots, t^0_m]$

Let n be the cardinal of $\{l^0_1, \ldots, l^0_a\}$. For each distinct l_i in $\{l^0_1, \ldots, l^0_a\}$ $(1 \leq i \leq n)$, we choose k in $\{1, \ldots, a\}$ such that $l_i = l^0_k$ and define $M_i = M^0_k$ and $t_i = l^0_k\sigma = l_i\sigma$. Besides, for every k' such that $l^0_{k'} = l^0_k$, we define $w_{k'} = \mathsf{w}_i$.

Let p be the cardinal of $\{y^0_1, \ldots, y^0_b\} \cap \mathrm{var}(l_1, \ldots, l_n)$. For each distinct y_j in $\{y^0_1, \ldots, y^0_b\}$ $(1 \leq j \leq p)$, we choose k in $\{1, \ldots, b\}$ such that $y_j = y^0_k$ and define $N_j = D_k[M^0_1, \ldots, M^0_m]$. Besides, for every k' such that $y^0_{k'} = y^0_k$, we define $w_{a+k'} = \mathsf{w}_{p+j}$.

Let $q = b - p$. We repeat the same operation for each distinct y_j in $\{y^0_1, \ldots, y^0_b\} - \mathrm{var}(l_1, \ldots, l_n)$ $(p+1 \leq j \leq p+q)$.

Finally, we let $D = D_0[w_1, \ldots, w_{a+b}]$. By construction, we have that

—$l = D[l_1, \ldots, l_n, y_1, \ldots y_{p+q}]$,

—the l_i are mutually distinct non-variable terms and the y_i are mutually distinct variables.

—$y_i \in \mathrm{var}(l_1, \ldots, l_n)$ iff $i \leq p$.

—$M_i \rhd t_i$ is in Φ,

—for every $1 \leq i \leq n$, $t_i = l_i\sigma$, and

—for every $1 \leq j \leq p+q$, $N_j \rhd_\Phi y_j\sigma$.

As for the last sentence, if D is a parameter, so is D_0. As $l = y^0_k$ is impossible for a convergent system $\mathcal{R}$, we have $D_0 = \mathsf{w}_k$ with $k \leq a$. Hence $C = \mathsf{w}_{j_k}$ and $t^0_k = C[t^0_1, \ldots, t^0_m] = l\sigma$ is not $\mathcal{R}$-reduced. □

Lemma 5.4 (completeness, context reduction). *Let (Φ, Ψ) be a state and M, t, t' be three terms such that $M \rhd_\Phi t$ and $t \to_\mathcal{R} t'$. Then, either $(\Phi, \Psi) \Longrightarrow^* \bot$ or there exist (Φ', Ψ'), M' and t'' such that*

—$(\Phi, \Psi) \Longrightarrow^* (\Phi', \Psi')$,

—$M' \rhd_{\Phi'} t''$ *with* $t' \to^*_\mathcal{R} t''$, *and*

—$\Psi' \models M \bowtie M'$.

*Besides, in both cases, the corresponding derivation from (Φ, Ψ) can be chosen to consist of a number of **B** rules, possibly followed by one instance of **A** rule involving the same rewrite rule $l \to r$ as the rewrite step $t \to_\mathcal{R} t'$.*

Proof. By hypothesis, there exist a (public) context C and some deduction facts $M^0_1 \rhd t^0_1, \ldots, M^0_{m_0} \rhd t^0_{m_0} \in \Phi$ such that $M = C[M^0_1, \ldots, M^0_{m_0}]$ and $t = C[t^0_1, \ldots, t^0_{m_0}]$.

Moreover, there exist a position α, a substitution σ and a rewrite rule $l \to r \in \mathcal{R}$ such that $t|_\alpha = l\sigma$ and $t' = t[r\sigma]_\alpha$.

We note that α must be a (symbol) position of C since the t^0_i are $\mathcal{R}$-reduced. Hence we may write $C|_\alpha[t^0_1, \ldots, t^0_{m_0}] = l\sigma$. We have that $M|_\alpha \rhd_\Phi l\sigma$.

By Lemma A.1, we deduce that there exist

—a proper (n, p, q)-decomposition D of l: $l = D[l_1, \ldots, l_n, y_1, \ldots y_p, z_1, \ldots z_q]$,

—$M_1 \rhd t_1$, …, $M_n \rhd t_n$ in Φ,

—N_1, …, N_{p+q}

such that

—for every $1 \leq i \leq n$, $t_i = l_i\sigma$,
—for every $1 \leq j \leq p$, $N_j \rhd_\Phi y_j\sigma$, and
—for every $1 \leq k \leq q$, $N_{p+k} \rhd_\Phi z_k\sigma$.

In particular, we obtain that

$$M|_\alpha = C|_\alpha[M^0_1, \ldots, M^0_{m_0}] \ \rhd_\Phi \ C|_\alpha[t^0_1, \ldots, t^0_{m_0}] = l\sigma$$
$$D[M_1, \ldots, M_n, N_1, \ldots, N_{p+q}] \ \rhd_\Phi \ D[t_1, \ldots, t_n, y_1\sigma, \ldots, y_p\sigma, z_1\sigma, \ldots, z_q\sigma] = l\sigma$$

Thus, by Lemma 5.3, there exists a derivation $(\Phi, \Psi) \Longrightarrow^* (\Phi_1, \Psi_1)$ using **B** rules such that $\Psi_1 \models M|_\alpha \bowtie D[M_1, \ldots, M_n, N_1, \ldots, N_{p+q}]$.

Besides, since y_j belongs to $\mathrm{var}(l_1, \ldots, l_n)$ by definition of decompositions, $y_j\sigma$ is a subterm of some $l_i\sigma = t_i$. Since $N_j \rhd_\Phi y_j\sigma$, by applying Lemma 5.2 repeatedly, we deduce that there exist some terms $M_{n+1}, \ldots, M_{n+p}$ and a derivation $(\Phi_1, \Psi_1) \Longrightarrow^* (\Phi_2, \Psi_2)$ using **B** rules such that for all j,

—$M_{n+j} \rhd y_j\sigma$ is in Φ_2, and
—$\Psi_2 \models M_{n+j} \bowtie N_j$.

Let $N = D[M_1, \ldots, M_{n+p}, N_{p+1}, \ldots, N_{p+q}]$. We deduce that $N \rhd_{\Phi_2} l\sigma$, and

$$\Psi_2 \models M|_\alpha \bowtie D[M_1, \ldots, M_n, N_1, \ldots, N_{p+q}] \bowtie N$$

We now consider the application to (Φ_2, Ψ_2) of a **A** rule that involves the rewrite rule $l \to r$, the decomposition D, the plain terms $(t_1, \ldots, t_{n+p}) = (l_1, \ldots, l_n, y_1, \ldots, y_p)\sigma$ and the substitution $\sigma' = \sigma|_V$ obtained by restricted the σ to the domain $V = \mathrm{var}(l_1, \ldots, l_n) = \mathrm{var}(l_1, \ldots, l_n, y_1, \ldots, y_p)$.

Case **A.3**. If $(r\sigma')\downarrow_\mathcal{R}$ is not ground and $\mathrm{Ctx}(\Phi_2^+ \vdash^?_\mathcal{R} r\sigma') = \bot$ where $\Phi_2^+ = \Phi_2 \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\}$, then we may conclude that $(\Phi_2, \Psi_2) \Longrightarrow \bot$ by an instance of rule **A.3** involving $l \to r$, the decomposition D and the facts $M_1 \rhd t_1, \ldots, M_{n+p} \rhd t_{n+p}$.

Case **A.1**. If there exists $N_0 = \mathrm{Ctx}(\Phi_2^+ \vdash^?_\mathcal{R} r\sigma')$ where $\Phi_2^+ = \Phi_2 \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\}$. By Property (b) of Ctx, let s_0 be such that $N_0 \rhd_{\Phi_2^+} s_0$ and $r\sigma' \to^*_\mathcal{R} s_0$, and define

—$\Phi' = \Phi_2$,
—$\Psi' = \Psi_2 \cup \{\forall z_1, \ldots, z_q.D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q] \bowtie N_0\}$,
—$M' = M[M_0]_\alpha$ where $M_0 = N_0\ \{z_i \mapsto N_{p+i}\}_{1 \leq i \leq q}$,
—$t'' = t[t_0]_\alpha = t'[t_0]_\alpha$ where $t_0 = s_0\ \{z_i \mapsto z_i\sigma\}_{1 \leq i \leq q}$.

By construction, we have $(\Phi_2, \Psi_2) \Longrightarrow (\Phi', \Psi')$ by an instance of rule **A.1**.

Besides, $r\sigma' \to^*_\mathcal{R} s_0$ implies $t'|_\alpha = r\sigma \to^*_\mathcal{R} t_0$ and $t' \to^*_\mathcal{R} t''$.

Given that $\alpha \in \mathrm{pos}(C)$ (where C is the previously context related to $M \rhd_\Phi t$) and $M_0 \rhd_{\Phi'} t_0$, we have that $M' = M[M_0]_\alpha \rhd_{\Phi'} t[t_0]_\alpha = t''$.

It remains to show that $\Psi' \models M \bowtie M'$. Indeed, we have seen that $\Psi_2 \models M|_\alpha \bowtie N$ where $N = D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q]\{z_i \mapsto N_{p+i}\}_{1 \leq i \leq q}$. Besides, by definition of Ψ', it holds that $\Psi' \supseteq \Psi_2 \supseteq \Psi_1$ and we have that $\Psi' \models D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q] \bowtie N_0$. Therefore, $\Psi' \models M|_\alpha \bowtie M_0$ and $\Psi' \models M \bowtie M[M_0]_\alpha = M'$.

Case **A.2**: if $(r\sigma')\downarrow_{\mathcal{R}}$ is ground and $\mathrm{Ctx}(\Phi_2^+ \vdash^?_{\mathcal{R}} r\sigma') = \bot$ where $\Phi_2^+ = \Phi_2 \cup \{z_1 \rhd z_1, \ldots, z_q \rhd z_q\}$, define

—$M_0 = D[M_1, \ldots, M_{n+p}, \mathsf{a}, \ldots, \mathsf{a}]$ and $t_0 = (r\sigma')\downarrow_{\mathcal{R}}$,
—$\Phi' = \Phi_2 \cup \{M_0 \rhd t_0\}$,
—$\Psi' = \Psi_2 \cup \{\forall z_1, \ldots, z_q.D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q] \bowtie M_0\}$,
—$M' = M[M_0]_\alpha$, and
—$t'' = t[t_0]_\alpha$.

where a is the fixed public constant of rule **A.2**.

By construction, $(\Phi_2, \Psi_2) \Longrightarrow (\Phi', \Psi')$ by an instance of the **A.2** rule.

Since t_0 is ground and $\sigma = \sigma'\sigma$, we have $t_0 = (r\sigma)\downarrow_{\mathcal{R}}$. Therefore $t' = t[r\sigma]_\alpha \to^*_{\mathcal{R}} t[\,(r\sigma)\downarrow_{\mathcal{R}}\,]_\alpha = t''$.

Given that $\alpha \in \mathrm{pos}(C)$ and by construction $M_0 \rhd_{\Phi'} t_0$, we have $M' \rhd_{\Phi'} t''$.

It remains to show that $\Psi' \models M \bowtie M'$. Indeed, we have seen that $\Psi_2 \models M|_\alpha \bowtie N$ where $N = D[M_1, \ldots, M_{n+p}, z_1, \ldots, z_q]\{z_i \mapsto N_{p+i}\}_{1 \le i \le q}$. By definition of Ψ', it holds that $\Psi' \models N \bowtie M_0$ hence $\Psi' \models M \bowtie M[N]_\alpha \bowtie M[M_0]_\alpha = M'$.

The additional properties claimed on the derivation are clear from the construction above. □

Received May 2010; revised May 2011; accepted January 2012