H. Comon−Lundh and V. Cortier

New decidability results for fragments of first−order logic and application to cryptographic protocols

Research Report LSV−03−2, Jan. 2003

Laboratoire Spécification et Vérification

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

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

Research Report LSV−03−2, Lab. Spécification et Vérification, CNRS & ENS de Cachan, France, Jan. 2003

This is the full version of the paper with same title in RTA'2003.

New decidability results for fragments of first-order logic and application to cryptographic protocols

Hubert Comon-Lundh and Véronique Cortier **

Laboratoire Spécification et Vérification, CNRS
Ecole Normale Supérieure de Cachan,
{comon,cortier}@lsv.ens-cachan.fr

Abstract. We consider a new extension of the Skolem class for first-order logic and prove its decidability by resolution techniques. We then extend this class including the built-in equational theory of exclusive or. Again, we prove the decidability of the class by resolution techniques.
Considering such fragments of first-order logic is motivated by the automatic verification of cryptographic protocols, for an arbitrary number of sessions; the first-order formalization is an approximation of the set of possible traces, for instance relaxing the nonce freshness assumption.
As an application of the decision results for extensions of the Skolem class, we get some new decidability results for the verification of cryptographic protocols without the perfect cryptography assumption: we may include the algebraic properties of exclusive or.
The proof of our main result relies on classical techniques: ordered strategies, narrowing modulo AC, semantic trees.

1 Introduction

The verification of cryptographic protocols deserved a lot of attention in the past few years, because of the huge application domain of secure communications via public channels. In this context, the full automation of verification tools is important because, in general, the same protocol appears in multiple contexts in a slightly altered form; each instance has to be verified since it is never clear whether a small modification has an impact on the security property or not.

Such verification problems are typically relevant to model checking: given a protocol P and a security property ϕ, does P satisfy ϕ ? And indeed, model-checking tools have been used successfully to find some attacks (the most famous one is due to G. Lowe [18]). However, *proving* the correctness of a protocol is much harder for several reasons. First of all, we must be very precise on the semantics of protocols and security properties; there is still today a debate on these aspects. Next, whatever model of the protocols is chosen, it is both infinite in depth (traces have an unbounded length, because arbitrarily many instances of the protocol, also called *sessions* can be involved) and infinitely branching (depending on an attacker's input). Finally most of the protocols use *nonces*, which are supposed to be randomly generated numbers. As demonstrated by several authors [8, 13, 1], this yields undecidability of model checking, even in very restricted cases.

** Partially supported by INRIA project SECSI and RNTL project EVA.

There are two possible research directions, which proved to be relevant: either consider a bounded number of sessions, which is sufficient to restore decidability as shown e.g. in [23], or to consider an abstraction of the model, which may be sufficient for proving the protocol correct, but may also output "dummy" attacks. This line of research is followed by e.g. [25, 3, 5] and that is also what we will consider in this paper.

A first abstraction consists in replacing nonces (randomly generated numbers) with terms depending on the context. That is what is done in all abstraction techniques we know. Then, protocols can be modeled within first-order logic [4, 25, 3] and the satisfaction of most popular security properties such as secrecy and authentication reduces to satisfiability of a set of clauses (see e.g. [11]).

However, even for protocols without nonces (or considering the above abstraction), the verification of simple properties remains undecidable (e.g. [8]). On the other hand, experiments using general purpose automatic theorem provers such as SPASS, show that, most of the time, the proof search terminates. Trying to explain this phenomenon reduces to finding decidable fragments of first-order logic in which most of the protocols can be expressed (with the above sketched abstraction for nonces). For instance, we have shown in [7] that, for a significant class of protocols, the confidentiality problem can be reduced to the solvability of a class of set constraints with equality, itself shown to be decidable using tree automata with memory.

On a completely different side, all automated verification results rely, so far, on the *perfect cryptography assumption*, which, roughly, says that the message algebra is a free term algebra. Such an hypothesis is too strong since many protocols use cryptographic primitives which do have algebraic properties. A typical example is the exclusive or. As an example, Bull's authentication protocol was proved to be secure with the perfect cryptography assumption, while there is an attack when the algebraic properties of `xor` are considered [21, 24]. Up to our knowledge, there are very few results on verification of cryptographic protocols with `xor`: the only other result is a proof of decidability in case of a bounded number of sessions [10].

The work described in this paper has two motivations: on one hand to explain the reasons why first-order theorem provers often terminate on protocol verification, on the other hand study the extensions considering the algebraic properties of exclusive or and an unbounded number of sessions.

We already realized in [7] that one reason for undecidability, which does not occur in practice, is the agents ability to copy and locally modify two distinct pieces of a message, hence enabling the simulation of two counters machines. That is why we will consider here protocols in which an agent can copy "blindly" at most one piece of the message he receives. "Blindly" has the following (informal) meaning: protocols consist in messages exchange between (say) two agents. Upon receiving some message m, agent A breaks m into pieces, decrypting what she can decrypt. Each piece she gets is either known to her (it can be a public value such as an agent name or a nonce she generated earlier,...) or something she does not know (a cyphertext that she cannot decrypt, a nonce generated by the other party). Such data are represented by variables: an intruder could for instance replace them by arbitrary values. If the message that A is supposed to send makes use of such variables, we say that she copies "blindly" their content.

Such an hypothesis on the uniqueness of blind copies seems relevant since most of the protocols of [6] falls into the class. On the model side, this corresponds to first-order clauses involving at most one variable. We will give more details in section 4. This gives however a first idea on why we consider here the fragment of first-order logic consisting in clauses which contain at most one variable. Actually, we have to consider a larger fragment, because we need to express for instance intruder capabilities, which do not fall in this category. More precisely, we consider a clausal fragment in which every clause C either contains at most one variable or is such that every subterm t is either ground, a variable, or contains all variables of the clause. We prove that this fragment of first-order logic is decidable in section 2, using ordered resolution techniques. This fragment is actually similar to the extension $\mathcal{S}^+$ of the Skolem class as defined in [15]. Still, it is different since for instance, we will allow literals $P(x)$ in multiple variables clauses. We also allow arbitrary ground literals.

Our main result is however the extension of this decidable class, considering the algebraic properties of `xor`: we prove in section 3 the decidability of fragment of first-order logic, which contains both the above class and the equality axioms for `xor`. For, we design a set of deduction rules and an ordered strategy, which we prove complete and terminating.

One difficulty here is that there is almost no ordering on terms with variables, which is stable by substitution. Hence we use an ordering which is stable by "non-collapsing" substitutions, restoring the completeness using a rule similar to narrowing. Another difficulty is to control the number of variables occurring in clauses (which we need for termination). To this end, we impose stronger restrictions on resolution and factorization, restoring completeness by adding in particular *extensions*.

Termination relies on technical results on unification with associativity, commutativity, identity and nilpotence (ACUN) and free symbols (which is known to be in NP [20]), typically concerning the sizes of mgus. Finally, completeness is obtained via classical semantic trees methods.

In section 4, we show how the previous results apply to the verification of cryptographic protocols, hence providing the first decidability result for an unbounded number of sessions, and considering the algebraic properties of `xor`. We illustrate the result, proving the correctness of a simple protocol. In [10], it is proved that protocol security is decidable in presence of `xor`, for a bounded number of sessions. Let us emphasize that we do not assume here a bounded number of sessions, however assuming some other properties of the protocol. The two results are actually disjoint and rely on completely different techniques: first-order logic is not relevant for a bounded number of sessions since it would require to give a bound on the number of times a clause is used (e.g. using rigid variables). On the other hand, in [10], there is no hypothesis on the number of blind copies and the result relies on constraint solving techniques and locality properties in the spirit of [19].

Due to space limitations, many proofs are only given in appendix.

2 A simple decidable fragment of first-order logic

2.1 Definitions

Let $\mathcal{F}$ be a finite set of function symbols, $\mathcal{V}$ a set of variables and $\mathcal{P}$ a finite set of predicates. For every clause C, $V(C)$ is the set of variables of C. If P is a positive literal, we write $L = \pm P$ for $L \in \{P, \neg P\}$. If u and t are terms of $\mathcal{T}(\mathcal{F} \cup \mathcal{V})$ and if x is a variable of u, $u[t/x]$ is the term u where every occurrence of x has been replaced by t.

Definition 1. *A clause set S belongs to the class $\mathcal{C}$ if for every clause C in S, either C contains at most one variable or, for every literal L in C :*

1. *either $L = \pm P(x_i)$ for some $P \in \mathcal{P}$;*
2. *or $L = \pm P(u[f(x_1, \ldots, x_n)/y])$ for some $P \in P$ and some $f \in \mathcal{F}$ such that $\{x_1, \ldots, x_n\} = V(C)$ and u is some term of $\mathcal{T}(\mathcal{F} \cup \{y\})$.*

We may write that a clause C is in $\mathcal{C}$ instead of saying that the set $\{C\}$ is in $\mathcal{C}$ to express that C is a clause of the form described above.

This class is incomparable with the class $\mathcal{S}^+$ as described in [15]. We believe that, with some additional technical details, we can extend our result so that our class contains $\mathcal{S}^+$. This is however not relevant for our application nor for the extension of the next section. As examples of sets of clauses that can be expressed in this class, let us mention for instance two-way alternating tree automata (see e.g. [9], chapter 7); since the emptiness of the automaton can also be expressed as a clause in the class, the decidability of $\mathcal{C}$ implies the emptiness decidability for two-way alternating automata.

If $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{V})$, $|t|$ is the *depth* of t (maximal size of its positions). For $x \in \mathcal{V}$, $|t|_x$ is the maximal depth of an occurrence of x in t. By convention, it is 0 if $x \notin V(t)$. $|.|$ and $|.|_x$ are extended to literals by $|P(t)| = |t|$ and $|P(t)|_x = |t|_x$.

We will prove the decision result by ordered resolution, using the ordering derived from the following definition.

Definition 2. *Let A, B be two literals.*

$$A < B \quad \text{if} \quad |A| < |B| \quad \text{and if} \quad \forall x \in V(A) \cup V(B) \quad |A|_x < |B|_x.$$

$A \leq B$ if $A < B$ or $A = B$.

Note that when $A \leq B$, we have in particular that $V(A) \subseteq V(B)$.

A sufficient condition for completeness of ordered resolution is to use a *liftable* ordering [17, 15], also called *stable* ordering in [16].

Definition 3 (liftability). *An ordering $\leq_{\mathcal{R}}$ is* liftable *if, for all atoms A, B and all substitutions θ, $A \leq_{\mathcal{R}} B$ implies $A\theta \leq_{\mathcal{R}} B\theta$.*

Proposition 1. *$\leq$ is a liftable ordering.*

2.2 Decidability result

Theorem 1 (decidability of $\mathcal{C}$). *Let S be a finite set of clauses such that S belongs to $\mathcal{C}$. The satisfiability of S is decidable.*

Proof sketch

We use splitting (see e.g. [26]), ordered factorization and ordered binary resolution (see e.g. [2]), w.r.t. the partial ordering defined above, using a classical redundancy criterion [2], also called *a posteriori criterion* in e.g. [15]; we apply resolution on two clauses C_1 and C_2 only if no atom of the resolvent is greater than the resolved atom. Such an ordered strategy is complete [2, 15]. It only remains to show termination.

First, after splitting, we only generate clauses in $\mathcal{C}$. Then, define $\|C\|$ as the maximal depth of its literals and let N be the maximum of $\|C\|$ for clauses in S. We show that, for every generated clause C' (after splitting), either $\|C'\| \leq N$, or else C' is ground and $\|C'\| \leq 2 \times N$. This is a consequence of simple lemmas on the unifiers of terms containing at most one variable, for instance:

Lemma 1. *Let u, v be two terms such that $V(u) = \{x\}$ and $V(v) = \{y\}$. If they are unifiable with mgu σ, then either $u\sigma$ is ground and $|u\sigma| \leq 2 \times \max(|u|, |v|)$ or else $|u\sigma| \leq \max(|u|, |v|)$.*

Then, thanks to the ordered strategy, which only unifies maximal literals, we get the bound on $\|C\|$ for the generated clauses C.

The termination follows from the fact that there are only finitely many clauses in $\mathcal{C}$ whose size is bounded and to which splitting does not apply.

3 An extension including the exclusive or

We are going to extend the result of the previous section, including algebraic properties of a binary symbol. We will proceed as in the previous section: we define an ordering and consider an ordered deduction strategy. There are however several additional problems:

- for termination purposes, we need to keep control on the number of variables in each clause. For, we restrict the applicability of e.g. resolution and restore completeness, adding extension rules.
- it is a hard task to find an ordering which is both liftable and compatible with the theory of `xor`. We use an ordering, which is stable only by substitutions which do not introduce any redex. Considering substitutions which introduce redexes is handled separately as a pre-processing step
- for termination purposes, we need to control the size of unifiers, relying on the particular equational theory we consider. We will see the analogs of lemma 1 in section 3.2.

3.1 Definition of the class of clauses

In this part, we extend our class of clauses $\mathcal{C}$ to a class of clauses $\mathcal{C}^{\oplus}$ including the algebraic properties of $\oplus$ which are described in figure 1. The two last equations can be oriented from left to right and we get a convergent rewrite system modulo associativity and commutativity, provided we add the extended rule $y \oplus x \oplus x \rightarrow y$ (see e.g. [12] for definitions). For any term t in $T(\mathcal{F} \cup \{\oplus\} \cup \mathcal{V})$, we write $t\downarrow$ its normal form w.r.t. these rules.

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

Fig. 1. Equational theory of the `xor` function symbol

Formally, we consider a finite set $\mathcal{F}$ of function symbols containing the constant symbol 0, a set $\mathcal{V}$ of variables and a finite set $\mathcal{P}$ of predicate symbols. $\mathcal{C}^{\oplus}$ is a class of clauses extending $\mathcal{C}$, described below.

Definition 4. *A clause set S belongs to $\mathcal{C}^{\oplus}$ if for every clause C in S, either C contains at most one variable or for every literal L in C :*

1. *either $L = \pm P(x_i)$ for some $P \in \mathcal{P}$;*
2. *or $L = \pm P(u[f(x_1, \dots, x_n)/y])$ for some $P \in \mathcal{P}$ and $f \in \mathcal{F}$ such that $\{x_1, \dots, x_n\} = V(C)$ and u is some term of $\mathcal{T}(\mathcal{F} \cup \{\oplus\} \cup \{y\})$;*
3. *or $C = \neg P(x_1) \vee \neg P(x_2) \vee P(x_1 \oplus x_2)$ for some $P \in \mathcal{P}$.*

Remarks : Note that for the second type of clauses $(\pm P(u[f(x_1, \dots, x_n)/y]))$, f is forbidden to be $\oplus$ but $\oplus$ may occur in u.

We will see in section 4 that the special clause $C_0 \overset{\mathrm{def}}{=} \neg I(x) \vee \neg I(y) \vee I(x \oplus y)$ is used to encode the ability of the intruder to compute the `xor` of two terms.

In the following, S_0 denotes the set of clauses in S which are of the third type in the above definition.

From now on, $=$ denotes the equality between terms (or literals) modulo the (AC) properties of the `xor` while $=_{\oplus}$ denotes the equality between terms (or literals) modulo the whole equational theory of the `xor`.

Following the AC property of $\oplus$, we assume terms written in flatten form: $\oplus$ may be considered as a variadic function symbol. Subterms are defined accordingly. For instance the subterms of $f(a \oplus b \oplus g(x))$ are $f(a \oplus b \oplus g(x)), a \oplus b \oplus g(x), a, b, g(x), x$. $a \oplus b$ and $a \oplus g(x)$ are not subterms.

We extend $|.|$ and $|.|_x$ on terms of $\mathcal{T}(\mathcal{F} \cup \{\oplus\} \cup \mathcal{V})$. Informally, since $\oplus$ is now a variadic symbol, it may in particular have only one argument, in which case we don't write it, hence don't count it in the size of the terms; that is why the following measure computes the length of the longest path, not taking $\oplus$ into account.

Definition 5. $\|.\|$ *is defined inductively by:*

1. $\|a\| = 1$ *if $a \in \mathcal{V}$ or if a is a constant symbol of $\mathcal{F}$;*
2. $\|f(t_1, \dots, t_k)\| = 1 + \max_{1 \leq i \leq k} \|t_i\|$ *for $f \in \mathcal{F}$;*
3. $\|t_1 \oplus \dots \oplus t_n\| = \max_{1 \leq i \leq n} \|t_i\|$ *if the head symbol of each t_i is not $\oplus$.*

Then $|t|$ is defined as $\|t \downarrow\|$. $|.|_x$ is defined in the same way except that $\|a\|_x = 1$ iff $a = x$. This is also extended to clauses by:

$$|C| = \max_{L \in C} |L| \quad \textit{and} \quad |\pm P(t)| = |t|.$$

Then the definition of $\leq$ (definition 2) is unchanged. However, $\leq$ is no longer a liftable ordering.

Example 1. Let $L_1 = P(a \oplus b)$, $L_2 = P(f(x \oplus a) \oplus f(b \oplus a))$ and $x\theta = b$. Then $L_1 < L_2$ but $L_1\theta = P(a \oplus b) \not< L_2\theta = P(0)$.

Actually, there are few orderings which are liftable and compatible with the rules of figure 1. For instance there is no such ordering which contains the subterm ordering: we would have $x \oplus f(a) > a$, but then $(a \oplus f(a)) \oplus f(a) > a$! That is why we introduce the notion of *narrow-liftable* ordering and *collapse-free* substitution.

Definition 6. *A substitution σ is* normalized *if, for every variable x, $x\sigma$ is in normal form. A substitution σ is* collapse-free *w.r.t. a set of terms S if, for every $t \in S$, $t\sigma \downarrow = t \downarrow \sigma$.*

We will write NS the set of normalized substitutions and $CF(C_1, \dots, C_n)$ the set of collapse-free substitutions w.r.t. the set of subterms occurring in the clauses $C_1, \dots, C_n$, which are supposed to be irreducible.

Definition 7. *An ordering $\leq_{\mathcal{R}}$ is* narrow-liftable *if, for every atoms A, B and every substitution θ, which is collapse-free w.r.t. B, $A <_{\mathcal{R}} B$ implies $A\theta <_{\mathcal{R}} B\theta$.*

Proposition 2. *$\leq$ is a narrow-liftable ordering on literals of clauses of $\mathcal{C}^{\oplus}$.*

3.2 Some useful results on unification

It is well known that unifiability modulo the theory of figure 1 is NP-complete in the presence of free function symbols and that unification is finitary [20]. We need however finer results (the analogs of lemma 1) to control the size of terms.

Lemma 2. *If $u \neq_{\oplus} v$ and $Var(u, v) \subseteq \{x\}$. Then either u and v are not unifiable (modulo the rules of figure 1) or else any (normalized) unifier $\sigma = \{x \mapsto w\}$ is such that w is a ground term and either w is a subterm of $u \oplus v$ or else $w = w_1 \oplus w_2$ is a normal form such that w_1 and $x \oplus w_2$ are subterms of u or v. Moreover, $|x\sigma| \leq \max\{|u|, |v|\}$.*

Note that $|u\sigma|$ may be strictly greater than $|u|$ and $|v|$.

Example 2. Let us consider $u = h^2(x) \oplus h^2(a) \oplus x$ and $v = h^2(x)$. The most general unifier of u and v is $\sigma(x) = h^2(a)$ and $u\sigma = h^4(a)$.

Lemma 3. *If $Var(u) \cap Var(v) = \emptyset$ and $Var(u) \subseteq \{x\}, Var(v) \subseteq \{y\}$, $u \neq_{\oplus} v$, then either u and v are not unifiable (modulo the rules of figure 1) or else every most general unifier θ of u, v is, up to variable renaming, such that:*

- *either there are ground subterms $w_1, \ldots, w_k$ of u, v such that $x\theta = w_1 \oplus \ldots \oplus w_k$ (resp. $y\theta = w_1 \oplus \ldots \oplus w_k$) and $y\theta$ is ground (resp. $x\theta$ is ground)*
- *or $x\theta = z \oplus t_1 \oplus \cdots \oplus t_k$, $y\theta = (u_1 \oplus \cdots \oplus u_n \oplus w_1 \oplus \cdots \oplus w_m)\theta$, where the t_i's and the w_i's are ground subterms of u, v, $n \geq 1$ and the u_i's are non-ground subterms of u or the converse, exchanging the roles of x and y (resp.of u and v).*

Example 3. $g(a \oplus f(y \oplus f(a)), f^n(y)) = g(x, f^n(x))$ has a solution $x = y = a \oplus f(a)$. Instantiating the original terms, their measure is growing.

A similar technique allows us to conclude when u or v is equal to $u'[x \to f(x_1, \ldots, x_n)]$. See the appendix for details.

We design the ordering $\leq$ in such a way that it is stable by collapse-free substitutions. Therefore, we have to show how it is possible to consider only such substitutions. A general result in [10] allows to focus on collapse-free substitutions, roughly guessing the shared parts and performing possible simplification beforehand. That is also what we (roughly) do here. However, we need also to control the size of the resulting clauses, taking advantage of our additional assumptions.

Lemma 4. *For every clause $C \in \mathcal{C}^{\oplus}$, there is a finite number of clauses $C_1, \ldots, C_n$ such that :*

$$\{C\sigma\downarrow \mid V(C\sigma) = \emptyset, \sigma \in NS\} = \bigcup_{i=1}^{n} \{C_i\sigma \mid V(C_i\sigma) = \emptyset, \sigma \in CF(C_1, \ldots, C_n)\}$$

Moreover, if $C \notin S_0$, every C_i falls in one of the three following cases: $C_i = C$, or C_i is ground and $|C_i| \leq 2 \times |C|$, or $V(C) = \{x\}$ and $C_i = C\{x' \mapsto y \oplus t_i\}\downarrow$ for some sum t_i of ground subterms of C.

3.3 The decidability result

The goal of this section is to prove the following (main) result:

Theorem 2 (decidability of $\mathcal{C}^{\oplus}$). *Let S be a finite set of clauses such that S belongs to $\mathcal{C}^{\oplus}$. The satisfiability of S is decidable.*

Thanks to lemma 4 we can restrict our attention to collapse-free substitutions, provided that we apply the rule which replaces C with the set of clauses C_i constructed in lemma 4. This rule is called *narrowing rule*.

But restricting ourself to "collapse-free" ordered resolution is still not sufficient to ensure termination. Indeed, only the repetitive resolution of renamings of C_0 with themselves yields an infinite set of clauses. That is why we will disallow resolution steps between clauses in S_0, restoring completeness using *extensions*. The situation is similar to the transitivity rule for which a special inference rule is designed: ordered chaining [2]. Extensions aim at inferring $P(s \oplus u) \vee C \vee D$ from $P(s \oplus t) \vee C$ and $P(t \oplus u) \vee D$ when t is maximal among s, t, u.

Deduction rules are displayed on figure 2. As usual (see e.g. [15]), repeatedly applying the deduction rules of figure 2 together with a splitting rule yields a set of sets of clauses: $\mathcal{S}_0 = \{S\}$ and $\mathcal{S}_{i+1}$ is obtained:

- either by replacing $S_j \in \mathcal{S}_i$ by $S_j \cup \{C\}$ if C can be inferred from S_i using a rule of figure 2,
- or by replacing some $S_j \cup \{C \vee C'\} \in \mathcal{S}_i$ with two sets $S_j \cup \{C\}$ and $S_j \cup \{C'\}$ if $Var(C) \cap Var(C') = \emptyset$.

We also remove redundant clauses at each step. For our purpose, it is sufficient to remove clauses $L \vee L \vee C$ when $L \vee C$ is in the set of clauses.

Lemma 5 (Correctness). *The narrowing rule and the deduction rules of figure 2 are correct (the set of models of one of the clause sets in $\mathcal{S}_i$ is the same as the set of models of one of the clause sets in $\mathcal{S}_{i+1}$) and, if every clause set in $\mathcal{S}_i$ is in $\mathcal{C}^{\oplus}$, then every clause set in $\mathcal{S}_{i+1}$ is in $\mathcal{C}^{\oplus}$.*

Lemma 6 (Termination). *The sequence $\mathcal{S}_i$ is finite when starting from $\mathcal{S}_0 = \{S\}$ and $S \in \mathcal{C}^{\oplus}$.*

Proof. (sketch) The sequence $\mathcal{S}_i$ is finite iff applying the rules of figure 2 together with the rule $C \vee C' \to C$ when $Var(C) \cap Var(C') = \emptyset$ terminates when starting from S.

We are going to give an upper bound on the size of a clause C in a set of $\mathcal{S}_i$. Let T the set of ground subterms of S and $N \overset{\text{def}}{=} \max_{L \in C, C \in S} |L|$. We show by induction on i that, for every clause C of a set of $\mathcal{S}_i$, either C is ground and $|C| \leq 2N$, or C is not ground and $|C| \leq N$, or C contains exactly one variable x and $|C\{x \mapsto x \oplus t\}| \leq N$ for some $t \in T$.

To prove this, we investigate all possible cases (each deduction rule) and we rely on lemma 2 and 3 (detailed proof in appendix D.2).

Then, we show that there are only finitely many ground clauses such that $|C| \leq 2N$ (this relies on the nilpotence of $\oplus$) and only finitely many non-ground clauses in $\mathcal{C}^{\oplus}$ such that $|C| \leq N$ or $|C\{x \mapsto x \oplus t\}| \leq N$ for some $t \in T$.

Binary Resolution

$$\frac{\neg P(t) \vee C \quad P(u) \vee C'}{C\sigma \vee C'\sigma}$$

If σ is collapse-free w.r.t. literals in C, C', $\sigma \in mgu(t, u)$, $P(t)\sigma \not< (C \vee C')\sigma$.

Factorization

$$\frac{L_1 \vee L_2 \vee C}{(L_1 \vee C)\sigma}$$

If $\sigma \in mgu(L_1, L_2)$, σ is collapse-free w.r.t. literals in the clause, and $L_1\sigma \not< C\sigma$.

Explosion

$$\frac{P(t \oplus u) \vee C}{(P(t \oplus u) \vee C)\sigma\downarrow}$$

If t is ground, $u\sigma$ is ground and $u\sigma < t$.

Extension 1

$$\frac{\neg P(t) \vee C \quad P(u_1 \oplus u_2) \vee C'}{(C \vee C' \vee \neg P(t_2 \oplus u_2))\theta} \quad \text{If} \left\{ \begin{array}{l} P(x \oplus y) \vee \neg P(x) \vee \neg P(y) \in S_0 \\ t = t_1 \oplus t_2 (\text{or } t = t_1 \text{ and } t_2 = 0) \\ Var(t) = Var(t_1), \theta \in mgu(t_1, u_1) \\ \theta \text{ collapse-free w.r.t. } t, u_1 \oplus u_2, t_2 \oplus u_2 \\ (C \vee C' \vee \neg P(t_2 \oplus u_2))\theta \not> t_1\theta. \end{array} \right.$$

Extension 2

$$\frac{P(t) \vee C \quad P(u_1 \oplus u_2) \vee C'}{(C \vee C' \vee P(t_2 \oplus u_2))\theta} \quad \text{If} \left\{ \begin{array}{l} \neg P(x) \vee \neg P(y) \vee P(x \oplus y) \in S_0 \\ t = t_1 \oplus t_2 (\text{or } t = t_1 \text{ and } t_2 = 0) \\ Var(t) = Var(t_1), \theta \in mgu(t_1, u_1) \\ \theta \text{ collapse-free w.r.t. } t, u_1 \oplus u_2, t_2 \oplus u_2 \\ (C \vee C' \vee \neg P(t_2 \oplus u_2))\theta \not> t_1\theta. \end{array} \right.$$

All rules only apply to non-splittable clauses, not belonging to S_0.

Fig. 2. Deduction rules.

Thanks to lemma 6, the sequence $\mathcal{S}_i$ is finite. We let $\mathcal{S}^*(S)$ be its limit, when starting from $\mathcal{S}_0 = \{S\}$, $S \in \mathcal{C}^{\oplus}$.

Lemma 7 (Completeness). *Let $S \in \mathcal{C}^{\oplus}$. S is unsatisfiable if and only if for every set $S' \in \mathcal{S}^*(S)$, $\bot \in S'$.*

Our deduction system is correct, thus if $\bot \in S^*$ for $S^* \in \mathcal{S}^*(S)$ then S is not satisfiable.

Assume S is not satisfiable and assume $\bot \notin S^* \in \mathcal{S}^*$.

We extend our partial ordering $\leq$ on literals to a total ordering $\tilde{\leq}$ on ground literals in the following way.

Let $\leq$ be any total ordering on the predicates $\mathcal{P}$ and on the function symbols $\mathcal{F}$. We extend $\leq$ on $\mathcal{F} \cup \{\oplus\}$ by $\oplus < f$ for all $f \in \mathcal{F}$. We let then $m(t)$ be the triple $(|t|, top(t), Sub(t))$ where $top(t)$ is the top symbol of t and $Sub(t)$ are its immediate (strict) subterms. For two ground terms in normal form, we let $t \tilde{<} t'$ if $m(t) < m(t')$ where the triples are lexicographically ordered, using the ordering on $\mathcal{F}$ for the second component, the lexicographic extension of $\tilde{\leq}$ on the subterms when the top symbol is not $\oplus$ and the multiset extension of $\tilde{\leq}$ otherwise.

Let L_1, L_2 be two ground positive literals : $L_1 = P_1(t_1)$ and $L_2 = P_2(t_2)$. Then $L_1 \tilde{<} L_2$ if either $t_1 \tilde{<} t_2$; or $t_1 = t_2$ and $P_1 < P_2$.

By definition, $\tilde{<}$ extends $<$ and $\tilde{\leq}$ is a total ordering. The Herbrand base is totally ordered accordingly as well as partial interpretations. As usual in semantic trees methods, since S is unsatisfiable, by correctness S^* is unsatisfiable, hence its semantic tree is finite (the set of partial interpretations which do not falsify a clause of S^*).

Then we consider a partial interpretation $\mathcal{I}$ whose two extensions to $P_1(v)$ falsify a clause of S^* and which is minimal w.r.t. the lexicographic ordering on partial interpretations. (This is a “leftmost” node whose two sons are failure nodes in the semantic tree). The lexicographic ordering on partial interpretations is defined by $I >_{lex} J$ if, when $P(u)$ is the maximal element of the Herbrand base such that I and J coincide on literals strictly smaller than $P(u)$, $I(P(u)) = 1$, $J(P(u)) = 0$.

By factorization we may assume that the two clauses C_1, C_2 falsified by the two extensions of $\mathcal{I}$ are such that $P_1(v) \vee C'_1 = C_1\sigma_1$ and $\neg P_1(v) \vee C'_2 = C_2\sigma_2$ for some $C_1, C_2 \in S^* \cup S_0$ such that $C'_1, C'_2 < P_1(v)$. By narrowing, we may assume that σ_1 is collapse-free w.r.t. C_1 and σ_2 is collapse-free w.r.t. C_2. We distinguish four cases: either $C_1, C_2 \in S_0$, or $C_1 \in S_0$ and $C_2 \notin S_0$, $C_1 \notin S_0$ and $C_2 \in S_0$ or else $C_1, C_2 \notin S_0$.

These cases are described in more details in the appendix, we sketch here the reasons why it works:

Case $C_1, C_2 \in S_0$: We prove directly that there is another smaller clause falsified by $\mathcal{I}$, simply recombining the terms in the right order. This corresponds to the uselessness of extensions of extensions.

Case $C_1 \in S_0, C_2 \notin S_0$: Let $C_1\sigma_1 = \neg P_0(x)\sigma_1 \vee \neg P_0(y)\sigma_1 \vee P_0(x \oplus y)\sigma_1$, $x\sigma_1 = v_1$, $y\sigma_1 = v_2$, and $(x\sigma_1 \oplus y\sigma_1)\downarrow = v$. There exist v'_1, v'_2, v' such that $v = v'_1 \oplus v'_2$, $v_1 = v'_1 \oplus v'$ and $v_2 = v'_2 \oplus v'$ without any collapse or $v = v_1 \oplus v_2$ without any collapse. We only consider the first case since the second one is similar. By hypothesis, $v_1 \tilde{<} v, v_2 \tilde{<} v$ and therefore $\mathcal{I}(P_0(v_1)) = \mathcal{I}(P_0(v_2)) = 1$. Assume w.l.o.g that $P_0(v_1) \tilde{\leq} P_0(v_2)$. Now, by minimality of the interpretation $\mathcal{I}$ (w.r.t.

lexicographic ordering), the partial interpretation $\mathcal{J}$ which coincides with $\mathcal{I}$ on literals strictly smaller than $P_0(v_1)$ and such that $\mathcal{J}(P_0(v_1)) = 0$ falsifies a clause $C_3 = P_0(u) \vee C'$ of S^*. We consider again two cases, depending on whether this clause is in S_0 or not.

Assume $C_3 \notin S_0$ and that no factorization can be applied. Also, by narrowing, $C_3\sigma_3$ does not contain any redex and $v_1 = u\sigma_3$. Moreover, $P_0(v_1)$ is maximal in $C_3\sigma_3$. We are going to show that we can apply **Extension 1** (possibly after **Explosion**) to C_2 and C_3 yielding a clause falsified by $\mathcal{I}$. We let $C_2 = \neg P_0(t) \vee C$. We have $v = t\sigma_2$ and σ_2 is collapse-free thus $t = t_1 \oplus t_2$ such that $t_1\sigma_2 = v_1'$ and $t_2\sigma_2 = v_2'$. In the same way, $u = u_1 \oplus u_2$ such that $u_1\sigma_3 = v_1'$ and $u_2\sigma_3 = v'$. This means in particular that t_1, u_1 are unifiable. By **Explosion**, we may assume that $V(t) = V(t_1)$ and, by lemma 3, that there is a $\theta \in mgu(t_1, u_1)$ such that $\sigma_2 \uplus \sigma_3 = \theta\theta'$. Moreover, let w be the maximal strict direct subterm of v. Since $v_2 \tilde{<} v_1 \tilde{<} v$, w is a strict direct subterm of v_1' thus $v_2 = v_2' \oplus v' \tilde{<} v_1'$. The inequality $v_2 = v_2' \oplus v' \tilde{<} v_1'$ gives $t_2\sigma_2 \oplus u_2\sigma_3 \tilde{<} t_1\sigma_2$, hence $(t_2 \oplus u_2)\theta\theta' \tilde{<} t_1\sigma_2$. It follows that $(t_2 \oplus u_2)\theta \not> t_1$. In addition, θ is collapse-free w.r.t. t, $t_1 \oplus t_2$, $t_2 \oplus u_2$ and the clauses C and C'. Then, we can apply **Extension 1** and there is a clause $(C \vee C' \vee \neg P_0(t_2 \oplus u_2))\theta$, which is already falsified by $\mathcal{I}$.

The case $C_3 \in S_0$ yields to the previous case where $C_1, C_2 \in S_0$.

Case $C_1 \notin S_0, C_2 \in S_0$: this case is symmetric to the previous one, replacing **Explosion 1** with **Explosion 2**.

Case $C_1, C_2 \notin S_0$. We simply use **Resolution**; there is a smaller clause which is already falsified by $\mathcal{I}$.

4 Application to cryptographic protocols

We assume the reader familiar with the notion of agent, nonce, intruder, ... In this paragraph, we show how security properties for a class of protocols can be expressed as the satisfiability of a set of clauses $S \in \mathcal{C}^{\oplus}$. We also propose a simple (new) cryptographic protocol, which we prove correct using our technique.

We have presented in [11] a clausal model of cryptographic protocols. This model is a generalization of Paulson's model [22] and the strand spaces model [14]. Unfortunately, it is much too expressive for decidability results. That is why we present here an abstraction of this model where the freshness of nonces is no longer guaranteed. This abstraction may induce false attack but is correct: if a protocol is proven correct in this model then it is correct in the general model.

Messages are terms constructed over the alphabet $\mathcal{F} = \{<\_,\_>, \{\_\}_\_, h\}$ and a finite set of constants C, depending on the protocol.

- $< m_1, m_2 >$ represents the concatenation of the two messages m_1 and m_2;
- $\{m_1\}_{m_2}$ represents the term m_1 encrypted by m_2;
- $h(m)$ represents the hash of m;

Note that we allow compound keys for example. We also could express asymmetric encryption but for the sake of simplicity, we do not present this in this paper. As explained in the introduction, this representation implicitly uses the perfect encryption assumption :

$$\{m\}_k = \{m'\}_{k'} \quad \Rightarrow \quad m = m' \;\&\; k = k'.$$

To relax this assumption, we add the $\oplus$ symbol together with its equational theory (described Fig. 1) : $m_1 \oplus m_2$ represents the message m_1 xored with the message m_2. The `xor` function is widely used to encrypt messages by block [6]. It can also be used to implement a computationally cheap encryption: if K is a private key, then, instead of encrypting m with K, we may simply xor m and K. This is the case in the Bull protocol described in [21]. We also propose the following protocol, which aims at sending a secret S_{ab}, shared by agents a, b, without using explicit encryption (hence using fewer time resources):

$$\begin{aligned} A \to B &: N_a \oplus K_{ab} \\ B \to A &: N_b \oplus N_a \\ A \to B &: S_{ab} \oplus N_b \end{aligned}$$

At the first step, the agent A sends a nonce N_a xored with the shared key between A and B. The protocol is designed in such a way that every xored message contains a random datum, hence preventing statistical attacks.

We consider a predicate I which represents the set of messages possibly known to the intruder. Abstracting nonces by constants, the first rule of our protocol can be represented by the following clause:

$$\Rightarrow I(n^1_{ab} \oplus K_{ab}), \tag{1}$$

where n^1_{ab}, K_{ab} are new function symbols. At the second step, the agent B can retrieve N_a by xoring the message he received by K_{ab}. Then he generates an new nonce N_b and sends the message $N_b \oplus N_a$. This can be represented by:

$$I(z) \Rightarrow I(z \oplus n^2_{ba} \oplus K_{ab}), \tag{2}$$

where n^2_{ba} is new function symbol. Eventually, when the agent A receives B's message, she can retrieve N_b and send a secret S_{ab} by xoring it with N_b:

$$I(z) \Rightarrow I(z \oplus n^1_{ab} \oplus S_{ab}). \tag{3}$$

These three clauses belong to our class $\mathcal{C}^{\oplus}$. Applying the reduction result of [11], we may assume that there are only two honest agents a, b and one dishonest agent c. We assume here that an honest agent is not allowed to speak with himself since we think this hypothesis is more realistic. Then, all clauses corresponding to the protocol rules are displayed figure 3. We use a finite set of constants $C = \bigcup_{i \in \{ab,ba,ac,ca,cb,bc\}} \{n^1_i, n^2_i, S_i\} \cup \{K_{ab}, K_{ac}, K_{bc}\}$.

Such a representation can be generalized to arbitrary protocols, and we stay within $\mathcal{C}^{\oplus}$ as soon as, at each step, at most one part of the message is blindly copied. Most of the protocols of [6] satisfy this property, like for example, the famous Needham-Schroeder public key protocol (and also its corrected version due to G. Lowe [18]).

It remains to describe the intruder capabilities: he sees every message sent through the network and may send new messages. He knows private keys of dishonest agents. In addition, he is able to compose and decompose messages. Intruder capabilities can be encoded by clauses of $\mathcal{C}^{\oplus}$. In particular, the ability of the intruder to apply the `xor` function is described by the clause $\neg I(x) \vee \neg I(y) \vee I(x \oplus y)$. Some of the clauses are described Fig. 4. Actually, only the three first rules are relevant for our example since we only use the $\oplus$ symbol.

First rule:

$$\begin{array}{lll} \Rightarrow I(n^1_{ab} \oplus K_{ab}) & \Rightarrow I(n^1_{ba} \oplus K_{ab}) & \Rightarrow I(n^1_{ac} \oplus K_{ac}) \\ \Rightarrow I(n^1_{ca} \oplus K_{ac}) & \Rightarrow I(n^1_{cb} \oplus K_{bc}) & \Rightarrow I(n^1_{bc} \oplus K_{bc}) \end{array}$$

Second rule:

$$\begin{array}{ll} I(z) \Rightarrow I(z \oplus n^2_{ab} \oplus K_{ab}) & I(z) \Rightarrow I(z \oplus n^2_{ba} \oplus K_{ab}) \\ I(z) \Rightarrow I(z \oplus n^2_{ac} \oplus K_{ac}) & I(z) \Rightarrow I(z \oplus n^2_{ca} \oplus K_{ac}) \\ I(z) \Rightarrow I(z \oplus n^2_{bc} \oplus K_{bc}) & I(z) \Rightarrow I(z \oplus n^2_{cb} \oplus K_{bc}) \end{array}$$

Third rule:

$$\begin{array}{ll} I(z) \Rightarrow I(z \oplus n^1_{ab} \oplus S_{ab}) & I(z) \Rightarrow I(z \oplus n^1_{ba} \oplus S_{ba}) \\ I(z) \Rightarrow I(z \oplus n^1_{ac} \oplus S_{ac}) & I(z) \Rightarrow I(z \oplus n^1_{ca} \oplus S_{ca}) \\ I(z) \Rightarrow I(z \oplus n^1_{bc} \oplus S_{bc}) & I(z) \Rightarrow I(z \oplus n^1_{cb} \oplus S_{cb}) \end{array}$$

Fig. 3. Rules representing our protocol for three participants a, b and c.

$\Rightarrow I(K_{ac})$ $\Rightarrow I(K_{bc})$	The intruder knows all keys of compromised agents.
$I(x), I(y) \Rightarrow I(x \oplus y)$	The intruder may apply the `xor` function to any messages.
$I(x), I(y) \Rightarrow I(\{x\}_y)$	The intruder can encrypt a known message with a known key.
$I(\{x\}_y), I(y) \Rightarrow I(x)$	The intruder can retrieve the clear text of a message encrypted with a known key.

Fig. 4. Some of the clauses defining I.

Now, the security property we want to ensure on this protocol is that the secret S_{ab} exchanged between the two honest agents a and b remains secret to the intruder. Such a property may be expressed by the clause: $\phi_0 \overset{\text{def}}{=} \neg I(S_{ab})$. Let $\mathcal{C}_P$ be the clauses described in Fig. 3 and Fig. 4. The protocol does not satisfy our security property if and only if $\mathcal{C}_P \cup \{\phi_0\}$ is not satisfiable: we are back to a satisfaction problem. Such a reduction to satisfiability actually holds for any purely negative security property [11].

As a consequence, the secrecy of our abstracted protocol can be decided by our decision procedure. And the answer is yes: our protocol preserves secrecy !

Proposition 3. *The set of clauses representing our protocol together with the security property $\mathcal{C}_P \cup \{\phi_0\}$ is satisfiable.*

Proof. We split the set of constants Γ into the set of (supposedly) secret data Γ_1 and known data Γ_2: $\Gamma_1 = \{n^1_{ab}, n^1_{ba}, n^2_{ab}, n^2_{ba}, S_{ab}, S_{ba}, K_{ab}\}$ and $\Gamma_2 = \Gamma \backslash \Gamma_1$. We consider a set of terms T (resp. T') such that an even (resp. odd) number of "secrets" data is xored:

$$T = \{u_1 \oplus \cdots \oplus u_n \oplus t_1 \oplus \cdots \oplus t_k \mid n \text{ is even, } u_i \in \Gamma_1, t_j \in \Gamma_2, u_i, t_j \text{ distinct}\}.$$

Then we consider the following set of clauses:

$$\begin{aligned} S^* \overset{\text{def}}{=} & \{I(m) \mid m \in T\} \cup \{\neg I(z \oplus m_1) \vee I(z \oplus m_2) \mid m_1 \oplus m_2 \in T\} \\ & \cup \{\neg I(m_1) \vee I(m_2) \mid m_1 \oplus m_2 \in T\} \cup \{\neg I(m) \mid m \in T'\}. \end{aligned}$$

S^* contains $\mathcal{C}_P \cup \{\phi_0\}$, thus it is sufficient to prove that S^* is satisfiable (actually S^* is obtained from $\mathcal{C}_P \cup \{\phi_0\}$ by applying our deduction rules thus S^* is satisfiable iff $\mathcal{C}_P \cup \{\phi_0\}$ is satisfiable). S^* is saturated by our inference rules (see appendix E). Applying theorem 2, since $\bot \notin S^*$, it follows that $\mathcal{C}_P \cup \{\phi_0\}$ is satisfiable. $\square$

Since the abstraction is an upper approximation, the above proposition shows that the protocol is secure.

Note: Instead of using the reduction result of [11], we could have introduced an arbitrary number of participants by adding new variables. For example, the second rule of our protocol could be represented by the clause:

$$A(x), A(y), I(z) \Rightarrow I(z \oplus n_2(x,y) \oplus K(x,y)),$$

where x nd y are variables representing agents. Such a clause does not belong to our class $\mathcal{C}^{\oplus}$ but we could extend $\mathcal{C}^{\oplus}$ to clauses with *basic variables* (like in [7]). Such basic variables may only represent restricted data like agents or nonces. We believe that the resulting class, which extends $\mathcal{C}^{\oplus}$, is still decidable.

5 Conclusion and perspectives

We have proved the decidability of a new first-order logic fragment, including some algebraic properties. This result applies to the automatic verification of cryptographic protocols.

There are few extensions to be considered: first, adding basic variables, as explained in the above note would be useful for the application. On the theoretical side, there is no reason to restrict the set S_0 in the definition of $\mathcal{C}^{\oplus}$ to a single predicate symbol: it should be possible to allow clauses such as $\neg P_1(x) \vee \neg P_2(y) \vee P_3(x \oplus y)$ where P_1, P_2, P_3 are distinct. We didn't consider this extension here for sake of simplicity and because we do not need it in the application.

Finally, the complexity of the decision result looks prohibitive. Before implementing the decision procedure, we need some refinements. First, we actually use a refinement of the ordering used in section 3: we established a general termination result, however, completeness holds for any ordering which is narrow liftable and which is compatible with the ordering used in the completeness proof on the ground level. In particular, we can use $\widetilde{\leq}$ on the ground level.

A last question is of course to get similar results for other equational theories. In this paper, however, we heavily rely on the particular theory of `xor`.

References

1. R. Amadio and W. Charatonik. On name generation and set-based analysis in the dolev-yao model. In *Proc. CONCUR 02*. Springer-Verlag, 2002.
2. L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 2. North Holland, 2001.
3. B. Blanchet. Abstracting Cryptographic Protocols by Prolog Rules (invited +talk). In P. Cousot, editor, *8th International Static Analysis Symposium (SAS'2001)*, volume 2126 of *Lecture Notes on Computer Science*, pages 433–436, Paris, France, July 2001. Springer Verlag.
4. D. Bolignano. Towards the mechanization of cryptographic protocol verificatio. In *Proc. 9th. In. onf on Computer Aided Verification*, number 1254 in Lecture Notes in Computer Science, 1997.

5. L. Bozga, Y. Lakhnech, and M. Périn. Abstract interpretation for secrecy using patterns. In *Proc. TACAS*, LNCS, 2003. To appear.
6. J. Clark and J. Jacob. A survey of authentication protocol literature: Version, 1997.
7. H. Comon and V. Cortier. Tree automata with one memory, set constraints and cryptographic protocols. Technical Report LSV-01-13, LSV, 2001. To appear in TCS.
8. H. Comon, V. Cortier, and J. Mitchell. Tree automata with memory, set constraints and ping pong protocols. In *Proc. ICALP 2001*, 2001.
9. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2002.
10. H. Comon and V. Shmatikov. Constraint solving, exclusive or and the decision of confidentiality for security protocols assuming a bounded number of sessions. Technical report, LSV, 2003.
11. H. Comon-Lundh and V. Cortier. Security properties: two agents are sufficient. In *Proc. European Symposium on Programming*, 2003. To appear. Also available as research report `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-10.rr.ps`.
12. N. Dershowitz. Rewriting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 9. North Holland, 2001.
13. N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on formal methods in security protocols*, 1999.
14. F. T. Fabrega, J. Herzog, and J. Guttman. Strand spaces: Proving security protocol correct. *Journal of Computer Security*, 7:191–230, 1999.
15. C. Fermuller, A. Leitsch, U. Hustadt, and T. Tamet. Resolution decision procedure. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 2, chapter 25, pages 1793–1849. North Holland, 2001.
16. J. Goubault-Larrecq and I. Mackie. *Proof Theory and Automated Deduction*, volume 6 of Applied Logic Series. Kluwer Academic, 1997.
17. R. Kowalski and P. Hayes. Semantic trees in automated theorem proving. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 87–101. Edinburgh University Press, 1969.
18. G. Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In Margaria and Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166, 1996.
19. D. McAllester. Automatic recognition of tractability in inference relations. *J. ACM*, 40(2):284–303, 1993.
20. P. Narendran, Q. Guo, and D. Wolfram. Unification and matching modulo nilpotence. In *Proc. CADE-13*, volume 1104 of *LNCS*, pages 261–274, 1996.
21. L. Paulson. Mechanized proofs for a recursive authentication protocol. In *Proc. 10th IEEE Computer Security Foundations Workshop*, pages 84–95, 1997.
22. L. C. Paulson. The Inductive Approach to Verifying Cryptographic Protocols. *Journal of Computer Security*, 6(1):85–128, 1998.
23. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc.14th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, June 2001.
24. P. Ryan and S. Schneider. An attack on a recursive authentication protocol: A cautionary tale. *Information Processing Letters*, 65(1):7–10, 1998.
25. C. Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In H. Ganzinger, editor, *Proc. 16th Conference on Automated Deduction*, volume 1632 of *Lecture Notes in Computer Science*, pages 314–328, 1999.
26. C. Weidenbach. Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 2, chapter 27. North Holland, 2001.

A Proofs of results in section 2

Proposition 1. *$\geq$ is a liftable ordering.*

Proof. Indeed, assume $A < B$. Let θ be any substitution. We have

$$|A\theta| = \max(|A|, \max_{x\in V(A)} (|A|_x + |\theta(x)|)).$$

Since $|A| < |B| < |B\theta|$ and $|A|_x < |B|_x$, we get $|A\theta| < |B\theta|$. Assume now that x is a variable of $A\theta$. Then

$$|A\theta|_x = \max(|A|_x, \max_{\substack{y \in V(A) \\ x \in V(\theta(y))}} |A|_y + |\theta(y)|_x),$$

thus $|A\theta|_x < |B\theta|_x$. □

Before proving lemma 1, we start with two additional lemmas.

Lemma 8. *If u, v are terms in $T(\mathcal{F} \cup \{x\})$ and $u \neq v$, then either u, v are not unifiable or they have a most general unifier $\sigma = \{x \mapsto t\}$ where t is a ground subterm of either u or v. Moreover $|u\sigma| \leq \max(|u|, |v|)$.*

Proof. (sketch) If u, v are distinct, there is a position p at which $u|_p \neq v|_p$. If for all such positions neither $u|_p$ nor $v|_p$ is a variable, then the two terms are not unifiable. Otherwise, $u|_p = x$ (resp. $v|_p = x$) and the mgu must be $\{x \mapsto v|_p\}$ (resp. $\{x \mapsto u|_p\}$).

Lemma 9. *Let u and v be two distinct terms with $V(u) \subseteq \{x\}$ and $V(v) \subseteq \{y\}$. If u and v are unifiable, then they have a mgu which falls in one of the following cases:*

1. *$x\sigma = y\sigma \in \{x, y\}$*
2. *$x\sigma = v'$, $y\sigma = y$ and v' is a subterm of v*
3. *$y\sigma = u'$, $x\sigma = x$ and u' is a subterm of u*
4. *$x\sigma$ and $y\sigma$ are ground and one of them is a subterm of u or v. Moreover, $|u\sigma| \leq 2 \times \max(|u|, |v|)$ in this case*

Proof. (sketch) Let us prove the lemma by induction on the sum of the sizes of u, v. In the base case, u or v is a variable and we fall into one of the cases 1,2,3. Remains the case where neither u nor v is a variable. If they are unifiable, they must have the same top symbol f: $u = f(u_1, \ldots, u_n)$ and $v = f(v_1, \ldots, v_n)$. Then a mgu of u, v is also a mgu of $u_1 = v_1 \wedge \ldots \wedge u_n = v_n$. Remove trivial equations from this conjunction (and let us keep the same notations). It remains at least one equation since $u \neq v$. Pick any of them $u_i = v_i$. By induction hypothesis, the mgu σ_i of u_i, v_i has one of the above four forms. Let us investigate these four cases:

case 1: In this case, applying the replacement $\{x \mapsto y\}$ to the remaining equations, we can apply lemma 8 to the remaining equations (if any). Then we fall into case 1 or case 4.

case 2: either $\sigma_i = \{x \mapsto v'_i\}$ is also a solution of the remaining equations and we fall into case 2 or else consider any other equation $u_j = v_j$ such that $v_j\sigma \neq u_j\sigma_i$. Applying the induction hypothesis to $u_j = v_j$, we cannot fall into cases 1,3 which would yield to a positive occur-check. If we are in case 4, switch i and j: then the case is considered below. Remains case 2: the mgu of $u_j = v_j$ is $\{x \mapsto v'_j\}$ where v'_j is a subterm of v_j. Now, any unifier of u, v is also a unifier of $v'_i = v'_j$. Then we apply lemma 8: the mgu of u, v is ground, $y\sigma$ is a subterm of v'_i or v'_j and $|x\sigma| = |v'_i\sigma| \leq |v_i|$. It follows that

$$|u\sigma| \leq |u| + |x\sigma| \leq |u| + |v_i| \leq |u| + |v| - 1 < 2 \times \max(|u|, |v|)$$

hence we fall into case 4.

case 3: it is similar to the previous one

case 4: σ_i must be a mgu of u, v since it is ground. The result follows from $|u_i| < |u|$ and $|v_i| < |u|$.

lemma 1. *Let u, v be two terms such that $V(u) = \{x\}$ and $V(v) = \{y\}$. If they are unifiable with mgu σ, then either $u\sigma$ is ground and $|u\sigma| \leq 2 \times \max(|u|, |v|)$ or else $|u\sigma| \leq \max(|u|, |v|)$.*

Proof. u, v are distinct since their variable sets are distinct. Then we use lemma 9:

$$|u\sigma| = |v\sigma| \leq \min(|u| + |x\sigma|, |v| + |y\sigma|)$$

and we get $|u\sigma| \leq \max(|u|, |v|)$ in the first three cases and $|u\sigma| \leq 2 \times \max(|u|, |v|)$ in the last case.

Lemma 9 can be reformulated in the cases where the two variables are (uniformly) replaced with terms $f(x_1, \dots, x_n)$:

Lemma 10. *Let $t_1 = u[f(x_1, \dots, x_k)/x]$ (resp. $V(t_1) = x$) and $t_2 = v[g(y_1, \dots, y_l)/y]$ be two distinct unifiable terms. Then, their mgu falls into one of the following cases:*

1. *either $k = l$ and for all $1 \leq i \leq k$, we have $\sigma(x_i) = \sigma(y_i) = z_i$;*
2. *or for all $1 \leq i \leq k$, $x_i\sigma = v_i[g(y_1, \dots, y_l)/y]$ (resp. $x\sigma = v_i[g(y_1, \dots, y_l)/y]$) and v_i is a subterm of v;*
3. *or for all $1 \leq j \leq l$, $y_j\sigma = u_i[f(x_1, \dots, x_k)]$ (resp. $y_j\sigma = u_i$) and u_i is a subterm of u;*
4. *or σ is ground and either the $x_i\sigma$ (resp. $x\sigma$) are subterms of u or v or the $y_j\sigma$ are subterms of u or v.*

In addition, in first three cases, $|t_1\sigma| = \max(|t_1|, |t_2|)$ and in the last case, $|t_1\sigma| = 2\max(|t_1|, |t_2|)$.

The size of the unified term may effectively increase.

Example 4. Consider $t_1 = f(f(x_1, x_2), h^n(f(x_1, x_2)))$ and

$$t_2 = f(f(h^n(a), h^n(a)), h^n(f(y_1, y_2))).$$

The substitution $\sigma = mgu(t_1, t_2)$ verifies $\sigma(x_1) = \sigma(y_1) = h^n(a)$ and $\sigma(x_2) = \sigma(y_2) = h^n(a)$. Thus $|t_1| = |t_2| = n + 3$ and $|t_1\sigma| = 2n + 3$.

Theorem 1. *Let S be a finite set of clauses such that S belongs to $\mathcal{C}$. The satisfiability of S is decidable.*

Proof. We use splitting (see e.g. [26]), ordered factorization and ordered binary resolution (see e.g. [2]), w.r.t. the partial ordering defined above, using a classical redundancy criterion [2], also called *a posteriori criterion* in e.g. [15]; we apply resolution on two clauses C_1 and C_2 only if no atom of the resolvent is greater than the resolved atom. Such an ordered strategy is complete [2, 15].

Then, define $\|C\|$ as the maximal depth of its literals. We prove first the following lemma

Lemma 11. *Let $C_1, C_2 \in \mathcal{C}$ and C be a clause obtained by ordered binary resolution on C_1, C_2. Then, for every clause C' obtained from C after splitting,*

- $C' \in \mathcal{C}$
- *If C' is not ground, then* $\|C'\| \leq \max(\|C_1\|, \|C_2\|)$
- *If C' is ground, then* $\|C'\| \leq 2 \times \max(\|C_1\|, \|C_2\|)$

Proof. (sketch) Let $C_1 = C_1' \vee L_1$, $C_2 = C_2' \vee \neg L_2$, $L_1 = \pm P(u)$, $L_2 = \mp P(v)$ and $C = (C_1' \vee C_2')\sigma$ where σ is a mgu of L_1, L_2.

We rely on lemmas 9 and 10: consider for instance the case in which $V(C_1) = \{x\}$ and $V(C_2) = \{y\}$. If σ is the identity (i.e. $u = v$ and the terms are ground), the result is straightforward (after splitting the resulting clause!) Otherwise, according to lemma 9, there are four cases:

In case 1 we simply get $C = C_1' \vee C_2'\{y \mapsto x\}$. C contains a single variable x , hence is in $\mathcal{C}$ and $\|C\| \leq \max(\|C_1\|, \|C_2\|)$

In case 2 $\sigma = \{x \mapsto v'\}$ and v' is a subterm of v. Then C contains the single variable y, hence belongs to $\mathcal{C}$. Moreover, $C = C_1'\sigma \vee C_2'$. To prove the inequalities on sizes, it is sufficient to consider the non-ground literals in C_1'. Let $L = \pm Q(t)$ be such a literal and suppose $|L\sigma| > |L_1\sigma|$. Consider a maximal length position p of x in t. Since $L_1\sigma$ is maximal, $|p| \leq |q|$ for some maximal length position q of x in u. Now, by lemma 1, $|u\sigma| \leq \max(|u|, |v|)$, hence $|q| + |x\sigma| \leq \max(|u|, |v|)$. It follows that $|t\sigma| \leq \max(|t|, |u|, |v|)$, hence $\|C_1'\sigma\| \leq \max(\|C_1\|, \|C_2\|)$.

Case 3 is similar to case 2

In case 4 , as in case 2, by maximality of L_1, for any position p of x in some literal $\pm Q(t)$ of C_1' (resp. C_2'), either $|Q(t)\sigma| \leq |L_1\sigma|$ or there is a position q of x in u (resp. a position q of y in v) such that $|p| \leq |q|$. Then

$$\begin{aligned}
|t\sigma| &\leq \max(|t|, |t|_x + |x\sigma|) \\
&\leq \max(|t|, |q| + |x\sigma|) \\
&\leq \max(|t|, |u\sigma|) \\
&\leq \max(|t|, 2 \times \max(|u|, |v|)) \\
&\leq 2 \times \max(\|C_1\|, \|C_2\|)
\end{aligned}$$

In the cases where C_1, C_2 contain more than one variable, the proofs are essentially the same as above, replacing the reference to lemma 9 with the reference to lemma 10:

Case 1 $L_1 = \pm P(x_1)$ and $L_2 = \mp P(x_2)$. Then, by maximality of $L_1\sigma$, all literals of C'_1 is of the form $\pm P_i(x_1)$ and all literals of C'_2 is of the form $\pm P_i(x_2)$. Thus $\sigma(x_1) = \sigma(x_2) = x$ and all literals of C is of the form $\pm P_j(x)$, thus $\|C\| = \|C_1\| = \|C_2\| = 1$.

Case 2 $L_1 = \pm P(x_1)$ and $L_2 = \mp P(u[f(y_1, \ldots, y_k)])$ (resp. $L_2 = \mp P(u(y))$). By maximality of $L_1\sigma$, all literals of C'_1 is of the form $\pm P_i(x_1)$. Thus $\sigma(x_1) = u[f(y_1, \ldots, y_k)]$ (resp. $\sigma(x_1) = u(y)$) which implies $C'_2\sigma = C'_2$ and for all literals L of C'_1, $L\sigma$ is of the form $\pm P_i(u[f(y_1, \ldots, y_k)])$ (resp. $\pm P_i(u(y))$). Thus C' is in $\mathcal{C}'$ and $\|C\| \leq \|C_2\|$.

Case 3 $L_1 = \pm P(u[f(x_1, \ldots, x_k)/z])$ and $L_2 = \mp P(v[g(y_1, \ldots, y_l)/z])$. Then, by lemma 10, there are three cases :

- either $k = l$ and for all $1 \leq i \leq k$, we have $\sigma(x_i) = \sigma(y_i) = z_i$;
- or for all $1 \leq i \leq k$, $\sigma(x_i) = v_i[g(y_1, \ldots, y_l)]$;
- or for all $1 \leq j \leq l$, $\sigma(y_j) = u_i[f(x_1, \ldots, x_k)]$.

In the first case, we get $C \in \mathcal{C}'$ and $\|C\| = \max(\|C_1\|, \|C_2\|)$. The second and third cases are equivalent, thus let us consider only the second case. $C'_2\sigma = C'_2$ and every literal of $C'_1\sigma$ is of the form $\pm P(w[g(y_1, \ldots, y_l)])$ thus C is still in $\mathcal{C}'$. For every literal L of C'_2, $|L\sigma| = |L| \leq \|C_2\|$. Let L be a literal of C'_1.

- Either $|L\sigma| \leq |L_1\sigma|$. Then, by lemma 10, $|L_1\sigma| \leq \max(|L_1|, |L_2|) \leq \max(\|C_1\|, \|C_2\|)$ thus $|L\sigma| \leq \max(\|C_1\|, \|C_2\|)$.
- Or $|L\sigma| > |L_1\sigma|$. Then, by maximality of $L_1\sigma$ in $C'_1\sigma$, there exists a variable y of $L_1\sigma$ such that $|L\sigma|_y \leq |L_1\sigma|_y$. Since the only variables of $L_1\sigma$ are the y_i, we get for all $1 \leq j \leq l$, $|L\sigma|_{y_j} \leq |L_1\sigma|_{y_j}$ which implies $L = \pm P_i(u'[f(x_1, \ldots, x_k)/z])$ with $|u'|_z \leq |u|_z$.

  Now, $|L\sigma| = |u'[f(x_1, \ldots, x_k)/z]\sigma| = \max(|u'|, |u'|_z + 1 + \max_{1 \leq i \leq k} |\sigma(x_i)|)$. Since $|u'| \leq |L| \leq \|C_1\|$ and $|u'|_z + 1 + \max_{1 \leq i \leq k} |\sigma(x_i)| \leq |L_1\sigma|$, we get $|L\sigma| \leq \max(\|C_1\|, \|C_2\|)$.

The other cases: $L_1 = \pm P(u[f(x_1, \ldots, x_k)/z])$ and $L_2 = \mp P(v(y))$ or $L_1 = \pm P(u(x))$ and $L_2 = \mp P(v(y))$ are similar to case 3. □

Proof of theorem 1 (continued): Now, let N be the maximum of $\|C\|$ for clauses in S. As a consequence of the above lemma, for every generated clause C' (after splitting), either $\|C'\| \leq N$, or else C' is ground and $\|C'\| \leq 2 \times N$.

On the other hand, there are only finitely many terms t in $T(\mathcal{F} \cup \{x\})$ such that $|t| \leq N$. This implies that there are finitely many terms of the form $t[f(x_1, \ldots, x_n)/y]$ with $t \in T(\mathcal{F} \cup \{y\})$ since $\mathcal{F}$ is finite. On the other hand, the number of variables in each clause of $\mathcal{C}$ is bounded (by definition). Hence, if we assume that no clause contains twice the same literal (we may assume w.l.o.g. that such clauses are eagerly replaced with clauses containing each literal only once), then there are only finitely many possible generated clauses.

B Narrow-liftability of the ordering

Proposition 2 *$\leq$ is a narrow-liftable ordering on literals of clauses of $\mathcal{C}^{\oplus}$.*

Proof. $|B\theta| = \|B\theta \downarrow\| = \|B \downarrow \theta\|$ and $\|A\theta \downarrow\| \leq \|A \downarrow \theta\|$, hence we may assume that A, B are in normal form. Then we prove, by induction on the depth of B that

$|B\theta| > |A\theta|$ and, for every $x \in Var(B), z \in Var(x\theta)$, $|B\theta|_z > |A\theta|_z$. In the base case, B is a constant or a variable x and we cannot have $|B| > |A|$. If $|A| < |B|$ and, for every variable $x \in Var(A, B)$, $|A|_x < |B|_x$, then let $A = f(t_1, \ldots, t_k)$, $B = g(u_1, \ldots, u_m)$ (f, g are possibly $\oplus$). By definition, $\max_{1 \leq i \leq k} |t_i| < \max_{1 \leq i \leq m} u_i$ and $\max_{1 \leq i \leq k} |t_i|_x < \max_{1 \leq i \leq m} |u_i|$. There are indices j_0 and j_x such that, for every i, $|t_i| < |u_{j_0}|$ and, for every x, $|t_i|_x < |u_{j_x}|_x$. Then we apply the induction hypothesis: for instance, for every i, $|t_i\theta|_z < |u_{j_x}\theta|_z$ for every variable $z \in Var(x\theta)$. It follows that $|A\theta|_z = \|A\theta \downarrow\|_z \leq \|A\theta\|_z < \|B\theta\|_z = |B\theta|_z$ since $B\theta$ is in normal form. We prove in the same way that $|A\theta| < |B\theta|$. □

C Useful results on unification

Lemma 2 *If $u \neq_\oplus v$ and $Var(u, v) \subseteq \{x\}$. Then either u and v are not unifiable or else any (normalized) unifier $\sigma = \{x \mapsto w\}$ is such that w is a ground term and either w is a subterm of $u \oplus v$ or else $w = w_1 \oplus w_2$ is a normal form such that w_1 and $x \oplus w_2$ are subterms of u or v. Moreover, $|x\sigma| \leq \max\{|u|, |v|\}$.*

Proof. (sketch) We let $s(u)$ be the *size* of u, i.e. the number of function symbols occurring in u (if a function symbol occurs n times, we count it n times). We prove by induction on $s(u) + s(v)$ that if $u \neq_\oplus v$ and θ is a normalized unifer of u, v, then either $x\theta$ is a subterm of (u or v) or else $x\theta = w_1 \oplus w_2$ (in normal form) such that w_1 and $x \oplus w_2$ are subterms of u or v.

If u and v are the variable x or constants, then the result holds.

Otherwise, $u = u_1 \oplus \ldots \oplus u_n$, $v = v_1 \oplus \cdots \oplus v_k$, where $u_1, \ldots, u_n, v_1, \ldots, v_k$ are not headed with $\oplus$. There are three cases. Either there exist i, j such that $u_i = v_j$ then the solutions of the equation $u = v$ are the solutions of the equation $u_1 \oplus \ldots u_{i-1} \oplus u_{i+1} \oplus \ldots \oplus u_n = v_1 \oplus \ldots v_{j-1} \oplus v_{j+1} \oplus \ldots \oplus v_k$, thus we apply the induction hypothesis (these two terms must be distinct since u, v are distinct).

Or none of the u_i, v_j is a variable. Then there exist i, j such that $u_i\theta \downarrow = u_j\theta \downarrow$ or $u_i\theta \downarrow = v_j\theta \downarrow$ or $v_i\theta \downarrow = v_j\theta \downarrow$. Then we may apply the induction hypothesis to u_i, u_j (resp. u_i, v_j, resp. v_i, v_j) except if $u = u_1 = f(u'_1, \ldots, u'_m)$ and $v = v_1 = f(v'_1, \ldots, v'_m)$. Then for every $1 \leq i \leq m$, $u'_i\theta \downarrow = v'_i\theta \downarrow$. Since $u \neq_\oplus v$, there is at least one index i such that $u'_i \neq_\oplus v'_i$. It suffices to apply the induction hypothesis to u'_i, v'_i.

Or there exists i such that $u_i = x$ but none of the v_j is a variable. Then $x\theta = (u'_1\theta \oplus \cdots u'_m\theta) \downarrow$ where $u'_i = u_j$ or v_j for some j. If there exists i, j such that $u'_i\theta \downarrow = u'_j\theta \downarrow$ then we simply apply the induction hypothesis to u'_i, u'_j which are distinct. If all u'_i's are ground, then there is a single solution $x = u'_1 \oplus \ldots \oplus u'_m$, which satisfies the desired properties.

Otherwise, assume u'_1 is not a ground term, and let $x \oplus t_1 \ldots \oplus t_k$ be an outermost occurrence of x in u'_1. Let $x\theta = w_1 \oplus \ldots \oplus w_m$ where the w_i's are distinct and not headed with $\oplus$. From $w_1 \oplus \ldots \oplus w_m = u'_1\theta \downarrow \oplus u'_2\theta \downarrow \ldots \oplus u'_n\theta \downarrow$, there must be some index j such that $w_j = u'_1\theta \downarrow$. w_j cannot occur in $u'_1\theta \downarrow$. Hence $w_j = t_i\theta \downarrow$ for some i. This means that θ is a solution of $t_i = u'_1$ and we may apply the induction hypothesis: $x\theta$ is either a subterm of t_i or a subterm of u'_1, or else $x\theta = w'_1 \oplus w'_2$ where w'_1 and $x \oplus w'_2$ are subterms of t_i or u'_1. The result follows since both t_i and u'_1 are subterms of u or v.

The last part of the lemma is a straightfoward consequence of the first part.

Lemma 3. *If $Var(u) \cap Var(v) = \emptyset$ and $Var(u) \subseteq \{x\}, Var(v) \subseteq \{y\}$, $u \neq_{\oplus} v$, then either u and v are not unifiable (modulo the rules of figure 1) or else every most general unifier θ of u, v is, up to variable renaming, such that:*

- *either there are ground subterms $w_1, \dots, w_k$ of u, v such that $x\theta = w_1 \oplus \dots \oplus w_k$ (resp. $y\theta = w_1 \oplus \dots \oplus w_k$) and $y\theta$ is ground (resp. $x\theta$ is ground)*
- *or $x\theta = z \oplus t_1 \oplus \dots \oplus t_k$, $y\theta = (u_1 \oplus \dots \oplus u_n \oplus w_1 \oplus \dots \oplus w_m)\theta$, where the t_i's and the w_i's are ground subterms of u, v, $n \geq 1$ and the u_i's are non-ground subterms of u or the converse, exchanging the roles of x and y (resp.of u and v).*

Proof. (sketch) Le θ be a most general unifier of u, v. We prove again the lemma by induction on $s(u) + s(v)$. First, if u or v is ground, the result follows from lemma 2: $x\theta$ (resp. $y\theta$) is either a ground subterm of u, v or the sum $w_1 \oplus w_2$ where w_1 is a ground subterm of u, v and w_2 is a sum of ground subterms of u, v.

If u and v are the variable x or constants, then the result holds.

Otherwise, $u = u_1 \oplus \dots \oplus u_n$, $v = v_1 \oplus \dots \oplus v_k$, where $u_1, \dots, u_n, v_1, \dots, v_k$ are not headed with $\oplus$. There are again four cases.

Case 1: $n = k = 1$. $u = u_1 = f(u'_1, \dots, u'_m)$ and $v = v_1 = f(v'_1, \dots, v'_m)$. Then we must have, for every i, $u'_i\theta\downarrow = v'_i\theta\downarrow$. Discard the indices i such that $u'_i = v'_i$. It remains at least one index j. We apply the induction hypothesis to $u'_j = v'_j$, considering a mgu σ of $u'_j = v'_j$ such that $\theta = \sigma\sigma'$ for some σ'. If σ is ground, then we must have $\sigma = \theta$, hence the desired results since ground subterms of u'_i, v'_j are also ground subterms of u, v. Otherwise, by symmetry, we may assume that $x\sigma = z \oplus t_1 \oplus \dots \oplus t_q$ and $y\sigma = (s_1 \oplus \dots \oplus s_p \oplus w_1 \oplus \dots \oplus w_r)\sigma\downarrow$ where $p \geq 1$, $s_1, \dots, s_p$ are non-ground subterms of u'_j and $t_1, \dots, t_k, w_1, \dots, w_r$ are ground subterms of u'_j, v'_j. If $u\sigma\downarrow = v\sigma\downarrow$, then the result is proved. Otherwise, σ' is a mgu of $u\sigma, v\sigma$. Then, we apply lemma 2: $\sigma' = \{z \mapsto w\}$ where either w is a ground subterm of $u\sigma, v\sigma$, or else $w = w'_1 \oplus w'_2$ where w'_1 is a ground subterm of $u\sigma, v\sigma$, w'_2 is ground and $z \oplus w'_2$ is a subterm of $u\sigma, v\sigma$. Note however that ground subterms of $u\sigma, v\sigma$ are necessary ground subterms of u, v since $x\sigma$ and $y\sigma$ are not ground and their ground subterms are also subterms of u, v. This implies that $x\theta = w \oplus t_1 \oplus \dots \oplus t_q$ is a sum of ground subterms of u, v (and $y\theta$ is ground) or else $x\theta = w'_1 \oplus w'_2 \oplus t_1 \oplus \dots \oplus t_q$ and w'_1 is a ground subterm of u, v and w'_2 is ground and $z \oplus w'_2$ is a subterm of $u\sigma, v\sigma$. This last property implies that w'_2 itself is a sum of ground subterms of u, v, hence $x\theta$ is also a sum of ground subterms of u, v.

Case 2: none of the u_i,v_j is a variable and $n + k > 2$. Then there exist i, j such that $u_i\theta\downarrow = u_j\theta\downarrow$ or $u_i\theta\downarrow = v_j\theta\downarrow$ or $v_i\theta\downarrow = v_j\theta\downarrow$. We remove u_i, u_j (resp. u_i, v_j, resp. v_i, v_j) from the sums $u_1 \oplus \dots \oplus u_n$ and $v_1 \oplus \dots \oplus v_k$, getting two terms u', v' such that $u'\theta\downarrow = v'\theta\downarrow$ and $s(u') + s(v') < s(u) + s(v)$. Either $u' = v'$ and we simply apply the induction hypothesis on the remaining pair (u_i, u_j or u_i, v_j or v_i, v_j) or else we can apply the induction hypothesis to u', v' and a most general unifier σ of u', v' such that $\theta = \sigma\sigma'$. Then, we proceed as in case 1.

Case 3: $u_i = x$ for some i and $v_1, \dots, v_k$ are not variables . If there exists l, j such that $u_l\theta\downarrow = u_j\theta\downarrow$ or $u_l\theta\downarrow = v_j\theta\downarrow$ or $v_l\theta = v_j\theta$, then, as in the previous cases, we can apply the induction hypothesis, yielding the desired result. This situation is now discarded. We rename the u_i's :

$$x\theta \oplus u_1\theta \downarrow \oplus \cdots \oplus u_n\theta \downarrow = v_1\theta \downarrow \oplus \cdots \oplus v_k\theta \downarrow .$$

There must be at least one v_j which is not ground since v is not ground. If all u_i's are ground, then there is a single (up to renaming) most general solution $x = u_1 \oplus \ldots \oplus u_n \oplus v_1 \oplus \cdots \oplus v_k$ and the result follows: $y\theta = z$, $x\theta = (u_1 \oplus \ldots \oplus u_n \oplus v_1 \ldots \oplus v_k)\theta$ which satisfies the property of the lemma.

Otherwise, assume u_1 is not a ground term, and let $x \oplus t_1 \ldots \oplus t_k$ be an outermost occurrence of x in u_1. Let $x\theta = w_1 \oplus \ldots \oplus w_m$ where the w_i's are distinct and not headed with $\oplus$. From

$$w_1 \oplus \ldots \oplus w_m = u_1\theta \downarrow \oplus u_2\theta \downarrow \ldots \oplus u_n\theta \downarrow \oplus v_1\theta \downarrow \oplus \ldots \oplus v_k\theta \downarrow,$$

there must be some index j such that $w_j = u_1\theta \downarrow$. w_j cannot occur in $u_1\theta \downarrow$. Hence $w_j = t_i\theta \downarrow$ for some i. This means that θ is a solution of $t_i = u_1$ and t_i is a strict subterm of u_1, hence distinct from u_1. Applying lemma 2, $x\theta$ is a sum of ground subterms of u, v and, replacing x with $x\theta$ in $u = v$, the new equation is non-trivial (y occurs in v), hence has only ground solutions. Then we meet the conclusions of the lemma.

Case 4: $u_i = x$ for some i, $v_j = y$ for some j. As before, if for some i, j $u_i\theta \downarrow = v_j\theta \downarrow$ (or $u_i\theta \downarrow = u_j\theta \downarrow$ or $v_i\theta \downarrow = v_j\theta \downarrow$), we conclude using the induction hypothesis. We assume that this does not occur.

We may rename the u_i's and the v_j's and we get

$$x\theta \oplus u_1\theta \downarrow \oplus \cdots \oplus u_n\theta \downarrow = y\theta \oplus v_1\theta \downarrow \oplus \cdots \oplus v_k\theta \downarrow .$$

If all u_i's are ground, there is a single unifier (up to renaming): $x\theta = u_1 \oplus \ldots \oplus u_n \oplus y \oplus v_1 \oplus \cdots \oplus v_k$, which meets the conditions of the lemma (with the identity on y). The same result holds if the v_j's are all ground.

Thus we may assume that u_1 and v_1 are not ground. Let $x\theta = w_1 \oplus \cdots \oplus w_l \oplus s_1 \oplus \cdots \oplus s_m$, $y\theta = w'_1 \oplus \cdots \oplus w'_{l'} \oplus s_1 \oplus \cdots \oplus s_m$ such that $w_i w'_i, s_i$ are not headed with $\oplus$, $l + l' = n + k$ and for each w_i, $w_i = u_j\theta \downarrow$ or $w_i = v_j\theta \downarrow$ f or some j (and the same property for the terms w'_i). Let $x \oplus t_1 \ldots \oplus t_k$ be an outermost occurrence of x in u_1 and $y \oplus t'_1 \oplus \cdots \oplus t'_{k'}$ be an outermost occurrence of y in v_1. If at least one of the w_i's is equal to some $t_j\theta \downarrow$, then $t_j\theta \downarrow = u_s\theta \downarrow$ or $v_s\theta \downarrow$ for some s, then we may apply the induction hypothesis to t_j, u_s (resp. t_j, v_s) and their unifier σ such that $\theta = \sigma\sigma'$ for some σ'. In case x is mapped to a sum of ground subterms of u, v and y is mapped to a ground term (or the converse),we get immediately the desired result since $\theta = \sigma$. Otherwise, either $u\sigma \downarrow = v\sigma \downarrow$ and we get again the conclusion or else σ' is a mgu of $u\sigma \downarrow, v\sigma \downarrow$ and, using a reasoning similar to that in case 1, we conclude that $z\sigma'$ is a sum of ground subterms of u, v and either $x\sigma$ or $y\sigma$ is also a sum of ground subterms of u, v.

The same reasoning applies if at least one of the w'_i is equal to some $t'_j\theta \downarrow$: then $t'_j\theta \downarrow = u_s\theta \downarrow$ or $v_s\theta \downarrow$ and we use the induction hypothesis.

We are left to the case where the w_i and $t_j\theta \downarrow$'s are all distinct and the w'_i's and $t'_j\theta \downarrow$'s are all distinct. Thus the w_i's are strict subterms of $u_1\theta \downarrow$ (since we have considered the outermost occurrence of x), which implies that $u_1\theta\downarrow$ is distinct from the w_i's, thus $u_1\theta\downarrow = w'_j$ for some j. Symmetrically, the w'_i's are strict subterms of $v_1\theta\downarrow$ and $v_1\theta\downarrow = w_m$ for some m. Thus we have:

$$w_m < u_1\theta\downarrow = w'_j < v_1\theta\downarrow = w_m,$$

which is a contradiction: the case can not happen, which completes the proof of the lemma.

Lemma 12. *If $Var(u) \cap Var(v) = \emptyset$ and $u = u'[x \to f(x_1, \ldots, x_n)]$, $Var(u') \subseteq \{x\}, Var(v) \subseteq \{y\}$, then either u and v are not unifiable (modulo the rules of figure 1) or else any most general unifier θ of u and v falls (up to renaming) in one of the following cases:*

- *for every i, $x_i\theta$ is a ground subterm of u, v and $y\theta$ is ground*
- *$y\theta$ is a sum of ground subterms of u, v and every $x_i\theta$ is ground*
- *$x_i\theta = x_i$ for every i and $y\theta = u_1 \oplus \cdots \oplus u_n \oplus w_1 \oplus \ldots \oplus w_m$ where every u_i is a non-variable, non ground subterm of u, $n \geq 1$, $w_1, \ldots, w_m$ are ground subterms of u, v.*
- *$y\theta = z \oplus t_1 \oplus \cdots \oplus t_k$, for every i, $x_i\theta = v_i\theta$ and $t_1, \ldots, t_k$ are ground subterms of u, v, $f(v_1, \ldots, v_n)$ is a subterm of v.*

If $Var(u) \cap Var(v) = \emptyset$ and $u = u'[x \to f(x_1, \ldots, x_n)]$, $v = v'[y \to g(y_1, \ldots, y_k)]$, $Var(u') \subseteq \{x\}, Var(v') \subseteq \{y\}$, then either u and v are not unifiable (modulo the rules of figure 1) or else any most general unifier θ of u and v falls (up to renaming) in one of the following cases:

- *for every i, j, $x_i\theta$ and $y_j\theta$ are ground subterms of u, v,*
- *$x_i\theta = x_i$ for every i and $y_j\theta = u_j$ for every j, such that $g(u_1, \ldots, u_k)$ is a subterm of u*
- *$y_i\theta = y_i$ for every i and $x_j\theta = v_j$ for every j, such that $f(v_1, \ldots, v_n)$ is a subterm of v.*

Proof. (sketch) Concerning the first part, the equation $u = v$ is equivalent to $u' = v \wedge x = f(x_1, \ldots, x_n)$. Then we apply lemma 3 to u', v and simplify the conclusions to meet the constraint that $x\theta$ is headed with f.

Concerning the second part of the lemma, $u = v$ is equivalent to $u = v' \wedge y = g(y_1, \ldots, y_k)$. Then we apply the first part of the lemma to u, v' and simplify the conclusions taking into account that $y\theta$ must be headed with g.

Lemma 4 *For every clause $C \in \mathcal{C}^{\oplus}$, there is a finite number of clauses $C_1, \ldots, C_n$ such that :*

$$\{C\sigma\downarrow \mid V(C\sigma) = \emptyset, \sigma \in NS\} = \bigcup_{i=1}^{n} \{C_i\sigma \mid V(C_i\sigma) = \emptyset, \sigma \in CF(C_1, \ldots, C_n)\}$$

Moreover, if $C \notin S_0$, every C_i falls in one of the following cases:

- *$C_i = C$*
- *C_i is ground and $|C_i| \leq 2 \times |C|$*

- $V(C) = \{x\}$ *and* $C_i = C\{x' \mapsto y \oplus t_i\} \downarrow$ *for some sum* t_i *of ground subterms of* C.

Proof. If $C \in S_0$, $C = \neg P(x) \vee \neg P(y) \vee P(x \oplus y)$, the clauses there are four clauses C_i: $C_1 = C$ itself, $C_2 \overset{\text{def}}{=} \neg P(x \oplus y) \vee \neg P(y) \vee P(x)$, $C_3 \overset{\text{def}}{=} \neg P(x) \vee \neg P(x \oplus y) \vee P(y)$ and

$$C_4 \overset{\text{def}}{=} \neg P(x \oplus z) \vee \neg P(y \oplus z) \vee P(x \oplus y).$$

Indeed, if we consider a normalized substitution σ such that $C\sigma \downarrow$ is ground, let $x\sigma \downarrow = t_1 \oplus \ldots \oplus t_k$, $y\sigma \downarrow = u_1 \oplus \ldots \oplus u_m$. If $\{t_1, \ldots, t_k\} \subseteq \{u_1, \ldots, u_m\}$ we get a collapse-free instance of C_3, if $\{u_1, \ldots, u_m\} \subseteq \{t_1, \ldots, t_k\}$, we get a collapse-free instance of C_2, if the two sets are disjoint, we get a collapse-free instance of C_1 and, in the general case, if $\{t_1, \ldots, t_k\} \cap \{u_1, \ldots, u_m\} = \{v_1, \ldots, v_n\}$ is non-empty and distinct from the two previous sets, we get a collapse-free instance of C_4 (for instance z is assigned $v_1 \oplus \ldots \oplus v_n$).

Consider now a clause C with only one variable x. Let T be the set of its ground subterms and $\sigma_1, \ldots, \sigma_n$ be the substitutions $\{x \mapsto x' \oplus t_1 \oplus \ldots \oplus t_m\}$ and $\{x \mapsto t_1 \oplus \ldots \oplus t_m\}$ where $\{t_1, \ldots, t_m\} \subseteq T$. We let then $C_i \overset{\text{def}}{=} C\sigma_i \downarrow$. Let σ be any normalized substitution. We proceed by induction on the number of reduction steps of $C\sigma$ to its normal form. If $C\sigma$ does not contain any redex, then it is a collapse-free instance of $C\{x \mapsto x'\}$ (with $m = 0$). Otherwise, consider an innermost redex: $u_1\sigma \oplus \ldots \oplus u_n\sigma$ in which $u_1\sigma = v_1 \oplus v_2$ and $u_2\sigma = v_1 \oplus v_3$ (v_2 and v_3 are possibly empty) and $u_1, \ldots, u_n$ are not headed with $\oplus$. If v_2 and v_3 are empty, since C is in normal form, σ must be a unifier of u_1 and u_2, hence, thanks to lemma 2, σ maps x to a sum of ground subterms of C, and $C\sigma \downarrow$ is equal to some C_i. Otherwise, since u_1, u_2 are not headed with $\oplus$, one of them must be a variable and, since C only contains one variable and is irreducible, we must have, e.g. v_3 empty and $u_1 = x$ (the other case is symmetric). Since we chose an innermost redex, u_2 must be a ground term (it cannot contain $u_2\sigma \oplus v_2$). Then we consider the substitution $\sigma_0 = \{x \mapsto u_2 \oplus x'\}$. $C\sigma \downarrow = (C\sigma_0) \downarrow \{x' \mapsto v_2\} \downarrow$ and $C\sigma_0 \downarrow \{x' \mapsto v_2\}$ contains strictly less redexes than $C\sigma$. We apply the induction hypothesis, replacing C with $C\sigma_0$ and σ with $\{x' \mapsto v_2\}$: there is a clause $C_i' = C\sigma_0 \downarrow \theta_i \downarrow$ and a collapse-free substitution τ_i such that $C_i'\tau_i = C\sigma_0 \downarrow \{x' \mapsto v_2\} \downarrow = C\sigma \downarrow$, where θ_i is a substitution $\{x' \mapsto y \oplus t\}$ or a substitution $\{x' \mapsto t\}$ with t a sum of ground subterms of $C\sigma_0 \downarrow$. Now, any ground subterm of $C\sigma_0 \downarrow$ is also a ground subterm of C, hence $t \in T$ and $\sigma_0\theta_i$ is one of the substitutions σ_j. It follows that $C_i' = C\sigma_j \downarrow = C_j$ and $C_j\theta_i = C\sigma \downarrow$, which completes the proof.

Consider finally the case where C contains more than one variable and does not belong to S_0: every literal is either $\pm P(x_i)$ or $L\{x \mapsto f(x_1, \ldots, x_n)\}$ where $V(L) \subseteq \{x\}$. We let C_i be the set of clauses containing C and the clauses obtained by unifying any two subterms u, v such that $u \oplus v$ occurs in C, and then normalizing. By lemma 2, the clauses C_i are ground clauses. Moreover, any redex in $C\sigma$ must be of the form $u\sigma \oplus v\sigma$ where $u \oplus v$ occurs in C. It follows that either $C\sigma$ is irreducible, or it is one of the clauses C_i.

It only remains to prove the size inequality for ground clauses C_i. When $V(C)$ contains only one variable, C_i is ground iff $C_i = C\{x \mapsto t\} \downarrow$ where t is a sum of ground subterms of C. Then $|C_i| \leq |C|_x + |t| \leq 2 \times |C|$. When C contains one variable, we simply consider literals others than $\pm P(x_i)$, abstract out $f(x_1, \ldots, x_n)$

with x and apply the same reasoning. The literal $\pm P(x_i)\sigma_i$ is smaller than other literals in the same clause.

D Proof of theorem 2

D.1 Correctness

Lemma 5 *The narrowing rule and the deduction rules of figure 2 are correct (the set of models of one of the clause sets in $\mathcal{S}_i$ is the same as the set of models of one of the clause sets in $\mathcal{S}_{i+1}$) and, if every clause set in $\mathcal{S}_i$ is in $\mathcal{C}^{\oplus}$, then every clause set in $\mathcal{S}_{i+1}$ is in $\mathcal{C}^{\oplus}$.*

Proof. The correctness of the rules of figure 2 is straightforward. For the narrowing rule, one can notice that, by construction, for every clause $C' \in N_C$, then $C' = C\sigma \downarrow$ for some σ. Let us focus on the $\mathcal{C}^{\oplus}$ membership: we proceed by induction on i. This true for $\mathcal{S}_0$ by definition and we show the invariance of this property by any deduction rule. According to the definition of $\mathcal{C}^{\oplus}$, we assume that every clause C is of one of the following four forms

1. C contains at most one variable;
2. C only contains literals $\pm P_i(x_i)$ where x_i are variables;
3. all maximal literals of C are of the form $\pm P_i(u)\{x \mapsto f(x_1, \dots, x_n)\}$ where $\{x_1, \dots, x_n\} = Var(C)$;
4. $C \in S_0$ but we do not apply resolution on these clauses.

In the second case, C is splittable or C falls into the first category.

Binary resolution: If the two clauses $C_1 = \neg P(t) \vee C$ and $C_2 = P(u) \vee C'$ contain at most one variable, then lemma 3 ensures that the resulting clause contains at most one variable.

If one of the two clauses contains more than one variable, then lemma 12 ensures that the resulting clause either contains at most contains one variable or is of the form $C\{x \mapsto f(x_1, \dots, x_n)\}$ where C contains one variable x, thus the resulting clause is in $\mathcal{C}^{\oplus}$.

Factorization and Explosion: These two rules leads to ground clauses which obviously in $\mathcal{C}^{\oplus}$.

Narrowing If C is a clause which contains at most one variable then the clauses obtained by narrowing also contain at most one variable thus are in $\mathcal{C}^{\oplus}$. If C is a clause of the form $C\{x \mapsto f(x_1, \dots, x_n)\}$ then the clauses obtained by narrowing are ground thus are in $\mathcal{C}^{\oplus}$.

Extensions As for binary resolution, the lemmas 3 and 12 allow us to conclude that the resulting clauses remain in $\mathcal{C}^{\oplus}$.

D.2 Termination

Lemma 6 *The sequence $\mathcal{S}_i$ must be finite when starting from $\mathcal{S}_0 = \{S\}$ and $S \in \mathcal{C}^{\oplus}$.*

Proof. (sketch)

The sequence $\mathcal{S}_i$ is finite iff applying the rules of figure 2 together with the rule $C \vee C' \to C$ when $Var(C) \cap Var(C') = \emptyset$ terminates when starting from S. We are going to give an upper bound on the size of a clause C in a set of $\mathcal{S}_i$. Let $N \overset{\text{def}}{=} \max_{L \in C, C \in S} |L|$ and let T be the set of sums of ground subterms of clauses of S.

We show by induction on i that, for every clause C of a set of $\mathcal{S}_i$, either C is ground and $|C| \leq 2N$ or C is not ground and $|C| \leq N$ or (last case) C is not ground, $V(C) = \{x\}$ and there exists $t \in T$ such that $|C\{x \mapsto x' \oplus t\} \downarrow | \leq N$.

If C' is obtained by narrowing from a clause C then the result follows from lemma 4.

If C' is obtained by **factorization** or **explosion** from a clause C, then C' is ground, $C' = C\sigma$ where σ is the most general unifier of two subterms of C. Applying lemma 2, $|\sigma| \leq N$. In addition, by induction there exists $t \in T$ such that $|C'''| \leq N$ and $C''' = C\{x \mapsto x' \oplus t\} \downarrow$ thus $|C'| = |C'''\{x \mapsto x'\sigma \oplus t\} \downarrow | \leq \max(|C'''|, |C'''| + \max(|t|, |\sigma|)) \leq 2N$.

If C' is obtained by **binary resolution**:

$$\frac{\neg P(t) \vee C \quad P(u) \vee C''}{C\sigma \vee C''\sigma},$$

$C' = C\sigma \vee C''\sigma$ and $\sigma \in mgu(t, u)$. The clauses $\neg P(t) \vee C$ and $P(u) \vee C''$ may be ground, have a single variable or be on the form $C_1[x \to f(x_1, \ldots, x_n)]$. All these cases are similar to case where $Var(t) = \{x\}$ and $Var(u) = \{y\}$, thus we only deal with that case. If σ is ground, then by lemma 2, $|\sigma| \leq N$ and we show similarly that $|C'| \leq 2N$. Otherwise σ is not ground and, by lemma 3, we may assume w.l.o.g. that $y\sigma = z \oplus t_1 \oplus \cdots \oplus t_k$ and $x\sigma = (u_1 \oplus \cdots \oplus u_m)\sigma \oplus t'_1 \oplus \cdots \oplus t'_k$, where the t_i's and the t'_i's are ground subterms of u or t and $m \leq 1$ and the u_i's are subterms of u. In addition, there exist $t_1, t_2 \in T$ (possibly 0) such that $P(t) \vee C = C_1\{x' \mapsto x \oplus t_1\} \downarrow$, $P(u) \vee C' = C_2\{y' \mapsto y \oplus t_2\} \downarrow$ and $|C_1| \leq N$, $|C_2| \leq N$.

We first prove the following lemma.

Lemma 13. *Let L_1, L_2 be two literals such that $Var(L_1) = Var(L_2) = \{x\}$ and L_1 and L_2 both belong to a clause C such that there exists t in T such that $|C\{x \mapsto x' \oplus t\} \downarrow | \leq N$. Let σ be a collapse-free substitution (w.r.t. L_1 and L_2) such that $L_1\sigma \not> L_2\sigma$ and $Var(x\sigma) = \{y\}$. Then $|L_1\sigma| \leq \max(N, |L_2\sigma|)$.*

In addition, let $\theta = \{y \mapsto x \oplus t\}$ be a substitution such that for each maximal length occurrence of y in $L_1\sigma$ and $L_2\sigma$, the variable y is xored by t. Then $|L_1\sigma\theta \downarrow | \leq \max(N, |L_2\sigma\theta \downarrow |)$.

Proof. Suppose $|L_1\sigma| > |L_2\sigma|$, then by maximality of $L_2\sigma$, we must have $|L_1\sigma|_y \leq |L_2\sigma|_y$ thus $|L_1|_x \leq |L_2|_x$. Then $|L_1\sigma| = |L_1|$ since for every position p of some $x \oplus u$ in L_1 (with u possibly equal to 0), $|p| + |x\sigma| \leq |L_2|_x + |x\sigma| \leq |L_2\sigma|$. Suppose also that $|L_1\sigma| > N$. It must be the case that there is a position p such that $L_1|_p = x \oplus u$ and $|L_1\sigma| = |L_1| = |p| + |u| > N$ since $|L_1\{x \mapsto x' \oplus t\} \downarrow | \leq N$. Let p' be a maximal position of the variable x in L_2: $L_2|_{p'} = x \oplus v$. Either $|v| \geq |u|$ in which case $|L_1\sigma| \leq |L_2\sigma|$, or $|v| < |u|$. Then since $|L_1\{x \mapsto x' \oplus t\} \downarrow | \leq N$, we must have $|t| = |u|$ thus $|L_2\{x \mapsto x' \oplus t\} \downarrow | \geq |L_2|_x + |t| > N$ (because $|v| < |t|$ implies

$|(v \oplus t)\downarrow| = |t|$), contradiction. We conclude that $|L_1\sigma| \leq N$, thus in any case, $|L_1\sigma| \leq \max(N, |L_2\sigma|)$.

Let us prove the second part of the lemma. We have $|L_1\sigma\theta\downarrow| \leq |L_1\sigma|$ and $|L_2\sigma\theta\downarrow| \leq |L_2\sigma|$. Assume $|L_1\sigma| > N$ and $|L_1\sigma\theta\downarrow| > N$ (if $|L_1\sigma| \leq N$ or $|L_1\sigma\theta\downarrow| \leq N$, we are done). If $|L_2\sigma\theta\downarrow| = |L_2\sigma|$ then we can conclude using the first part of the lemma. Thus consider the case where $|L_2\sigma\theta\downarrow| < |L_2\sigma|$. This means that $|L_2\sigma| = |L_2\sigma|_y + |t|$. Suppose $|L_1\sigma|_y > |L_2\sigma|_y$ then $|L_1\sigma| \geq |L_1\sigma|_y + |t| > |L_2\sigma|_y + |t| = |L_2\sigma|$, which contradicts the maximality of $L_2\sigma$. Thus $|L_1\sigma|_y \leq |L_2\sigma|_y$. Let p be a path such that $L_2\sigma|_p = y \oplus u$ and $|p| = |L_2\sigma|_y$. Now two cases are possible:

- Either $|L_1\sigma\theta\downarrow|$ is reached for some path extending a position of x in L_1, then we conclude using $|L_1\sigma|_y \leq |L_2\sigma|_y$ that $|L_1\sigma\theta\downarrow| \leq |L_2\sigma\theta\downarrow|$.
- Or $|L_1\sigma\theta\downarrow|$ is reached for some path p' such that $L_1\sigma|_{p'} = y \oplus v$ and $|p'| + |v| > N$. We may assume that p' does not extend a position x in L_1 otherwise we are back to the previous case. Thus $L_1|_{p'} = x \oplus v'$, i.e. $L_1\sigma|_{p'} = x\sigma \oplus v' = y \oplus v_1 \oplus v'$ with $v = v_1 \oplus v'$ and $x\sigma = y \oplus v_1$. If $|v_1| \geq |v'|$, we are back to the previous case. Otherwise $|v_1| < |v'|$. Since $|L_2\{x \mapsto x' \oplus t\}\downarrow| \leq N$, we must have $|u| = |v'|$, thus $|L_1\sigma\theta\downarrow| = |p'| + |v| \leq |p| + |u| \leq |L_2| \leq L_2\sigma\theta\downarrow$. □

From this lemma and the fact that $L\sigma \not> P(t)\sigma$ and $L'\sigma \not> P(u)\sigma$, we deduce that $|L\sigma| \leq \max(N, |P(t)\sigma|)$ for every literal L of C and $|L'\sigma| \leq \max(N, |P(u)\sigma|)$ for every literal L' of C''. Now $P(t)\sigma = P(u)\sigma = P(u)\{y \mapsto z \oplus t_1 \oplus \cdots \oplus t_k\}$ and since the clauses are in normal form, the t_i's occur at each maximal occurrence of the variable z in $C\sigma \vee C''\sigma$. Let $z\theta = y \oplus t_1 \oplus \cdots \oplus t_k$. We deduce from lemma 13 that $|L\sigma\theta\downarrow| \leq \max(N, |P(t)\sigma\theta\downarrow|)$ for every literal L of C and $|L'\sigma\theta\downarrow| \leq \max(N, |P(u)\sigma\theta\downarrow|)$ for every literal L' of C''. Note that actually, $P(u)\sigma\theta\downarrow = P(u)$. Suppose $|P(u)| > N$. Since $P(u) \vee C' = C_2\{x \mapsto x \oplus t_2\}\downarrow$ with $|C_2| \leq N$, this means that there exists a term t'_2 such that t'_2 occurs at each maximal occurrence of the variable y in C'' thus also occurs at each maximal occurrence of the variable y in $C\sigma\theta\downarrow$ and such that $P(u) \vee C' = C'_2\{x \mapsto x \oplus t'_2\}\downarrow$ with $|C'_2| \leq N$. Let $y\theta = y \oplus t'_2$. Applying again lemma 13, we get $|L\sigma\theta\theta'\downarrow| \leq \max(N, |P(t)\sigma\theta\theta'\downarrow|)$ for every literal L of C and $|L'\sigma\theta\theta'\downarrow| \leq \max(N, |P(u)\sigma\theta\theta'\downarrow|)$ for every literal L' of C''. Since $|P(u)\sigma\theta\theta'\downarrow| \leq N$ and $P(t)\sigma\theta\theta'\downarrow = P(u)\sigma\theta\theta'\downarrow$, we have that $C' = C\sigma \vee C''\sigma = C_3\theta\theta'$ with $|C_3| \leq N$, hence the result.

If C' is obtained by one of the **Extension** rules, the proof is similar to the binary resolution.

Now, for each non ground clause C, there is a term $t \in T$ (possibly 0) such that $|C\{x \mapsto x' \oplus t\}\downarrow| \leq N$ thus $|C| \leq 2N$. Thus for every clause C (ground or not) in $\mathcal{S}_i$, we have $|C| \leq 2N$. Moreover, there are only finitely many clauses C, which do not contain repeated literals and such that $|C| \leq 2N$. Indeed, there are finitely many literals L in normal form such that $|L| \leq 2N$ and containing at most one variable.

D.3 Completeness

Lemma 14 (Case $C_1, C_2 \in S_0$). *If $C_1, C_2 \in S_0$, then $\mathcal{I}$ falsifies already a ground instance of $S^* \cup S_0$.*

Notation : If $t = t_1 \oplus \cdots \oplus t_k$ where the head symbol of each t_i is not the `xor` symbol, then $\widetilde{t} \overset{\text{def}}{=} \{t_1, \ldots, t_k\}$. If $t = f(t_1, \ldots, t_n)$, then $\widetilde{t} \overset{\text{def}}{=} \{t\}$.

Proposition 4. *Let t, t_1, t_2 be three ground terms in normal form such that $t = t_1 \oplus t_2$ and $t_1, t_2 \tilde{<} t$. Let u be the maximal term of $\widetilde{t}$. Then $u \notin \widetilde{t_1}$ or $u \notin \widetilde{t_2}$.*

Proof. Assume $u \in \widetilde{t_1}$ and $u \in \widetilde{t_2}$, then $u \notin \widetilde{t_1 \oplus t_2} = \widetilde{t}$, contradiction. □

Proof (of lemma 14). $C_1\sigma_1 = P(v) \vee \neg P(v_1) \vee \neg P(v_2)$, $v_1, v_2 \tilde{<} v$ $C_2\sigma_2 = \neg P(v) \vee P(v_3) \vee \neg P(v_4)$, $v_3, v_4 \tilde{<} v$ and $\neg P(v_1) \vee \neg P(v_2)$ and $P(v_3) \vee \neg P(v_4)$ are already falsified.
Let u be the maximal term of $\widetilde{v}$, using proposition 4, $u \notin \widetilde{v_1}$ or $u \notin \widetilde{v_2}$ and $u \notin \widetilde{v_3}$ or $u \notin \widetilde{v_4}$. v_1 and v_2 play symmetric roles, thus we may assume w.l.o.g that $u \notin \widetilde{v_2}$.

- Assume $u \notin \widetilde{v_3}$. Then $v_2 \oplus v_3 \tilde{<} u \tilde{<} v$, thus $P(v_2 \oplus v_3)$ is already interpreted.
  - Either $P(v_2 \oplus v_3) \in \mathcal{I}$, then the clause $C = P(v_3) \vee \neg P(v_2) \vee \neg P(v_2 \oplus v_3)$ is an instance of of a clause of S_0, falsified by $\mathcal{I}$.
  - Or $\neg P(v_2 \oplus v_3) \in \mathcal{I}$. $v_2 \oplus v_3 = v \oplus v_1 \oplus v \oplus v_4 = v_1 \oplus v_4$. Thus the clause $C = \neg P(v_1) \vee \neg P(v_4) \vee P(v_2 \oplus v_3)$ (instance of a clause of S_0) is falsified by $\mathcal{I}$.
- Assume $u \notin \widetilde{v_4}$. Then $v_2 \oplus v_4 \tilde{<} u \tilde{<} v$, thus $P(v_2 \oplus v_4)$ is already interpreted. We proceed similarly.
  - Either $\neg P(v_2 \oplus v_4) \in \mathcal{I}$, then the clause $C = \neg P(v_2) \vee \neg P(v_4) \vee P(v_2 \oplus v_4)$ is an instance of C_0, falsified by $\mathcal{I}$.
  - Or $P(v_2 \oplus v_4) \in \mathcal{I}$. $v_2 \oplus v_4 = v \oplus v_1 \oplus v \oplus v_3 = v_1 \oplus v_3$. Thus the clause $C = \neg P(v_1) \vee P(v_3) \vee \neg P(v_2 \oplus v_4)$ (instance of a clause of S_0) is falsified by $\mathcal{I}$. □

Lemma 15 (Case $C_1 \in S_0$, $C_2 \notin S_0$). *If $C_1 \in S_0$, $C_2 \notin S_0$, then $\mathcal{I}$ falsifies already a ground instance of some clause in $S^* \cup S_0$.*

Proof. Let $C_1\sigma_1 = \neg P(x)\sigma_1 \vee \neg P(y)\sigma_1 \vee P(x \oplus y)\sigma_1$, $x\sigma_1 = v_1$, $y\sigma_1 = v_2$, and $(x\sigma_1 \oplus y\sigma_1) \downarrow = v$. There exist v_1', v_2', v' such that $v = v_1' \oplus v_2'$, $v_1 = v_1' \oplus v'$ and $v_2 = v_2' \oplus v'$ without any collapse or $v = v_1 \oplus v_2$ without any collapse. We only consider the first case since the second one is similar. By hypothesis, $v_1 \tilde{<} v, v_2 \tilde{<} v$ and therefore $\mathcal{I}(P(v_1)) = \mathcal{I}(P(v_2)) = 1$. Assume w.l.o.g that $P(v_1) \tilde{\leq} P(v_2)$. Now, by minimality of the interpretation $\mathcal{I}$ (w.r.t. lexicographic ordering), the partial interpretation $\mathcal{J}$ which coincides with $\mathcal{I}$ on literals strictly smaller than $P(v_1)$ and such that $\mathcal{J}(P(v_1)) = 0$ falsifies a clause $C_3 = P(u) \vee C'$ of S^*. We consider again two cases, depending on whether this clause is in S_0 or not.

Assume $C_3 \notin S_0$ and that no factorization can be applied. Also, by narrowing, $C_3\sigma_3$ does not contain any redex and $v_1 = u\sigma_3$. Moreover, $P(v_1)$ is maximal in $C_3\sigma_3$. We are going to show that we can apply **Extension 1** (possibly after **Explosion**) to C_2 and C_3 yielding a clause falsified by $\mathcal{I}$. We let $C_2 = \neg P(t) \vee C$. We have $v = t\sigma_2$ and σ_2 is collapse-free thus $t = t_1 \oplus t_2$ such that $t_1\sigma_2 = v_1'$ and $t_2\sigma_2 = v_2'$. In the same way, $u = u_1 \oplus u_2$ such that $u_1\sigma_3 = v_1'$ and $u_2\sigma_3 = v'$. This means in particular that t_1, u_1 are unifiable. By **Explosion**, we may assume that $V(t) = V(t_1)$ and, by lemma 3, that there is a $\theta \in mgu(t_1, u_1)$ such that $\sigma_2 \uplus \sigma_3 = \theta\theta'$. Moreover, let w be the maximal term of $\widetilde{v}$. Since $v_2 \tilde{<} v_1 \tilde{<} v$, w is a term of $\widetilde{v_1'}$ thus $v_2 = v_2' \oplus v' \tilde{<} v_1'$. The inequality $v_2 = v_2' \oplus v' \tilde{<} v_1'$ gives $t_2\sigma_2 \oplus u_2\sigma_3 \tilde{<} t_1\sigma_2$, hence $(t_2 \oplus u_2)\theta\theta' \tilde{<} t_1\sigma_2$. It follows that $(t_2 \oplus u_2)\theta \not> t_1$. In addition, θ is collapse-free w.r.t. t, $t_1 \oplus t_2$, $t_2 \oplus u_2$ and the clauses C and C'. Then, we can apply **Extension 1** and there is a clause $(C \vee C' \vee \neg P(t_2 \oplus u_2))\theta$, which is already falsified by $\mathcal{I}$.

Now, if $C_3 \in S_0$. Then $C_3 = P(x' \oplus y) \vee \neg P(x') \vee \neg P(y')$ Let $x'\sigma_3 = w_1$, $y'\sigma_3 = w_2$, and $(x'\sigma_3 \oplus y'\sigma_3) \downarrow = v_1$. We have $w_1 \tilde{<} v_1$ and $w_2 \tilde{<} v_1$. We may assume $w_2 \tilde{<} w_1$. In addition, $\mathcal{I}(P(w_1)) = 1$ and $\mathcal{I}(P(w_2)) = 1$. Let us consider the term $w_2 \oplus v_2$. We have $w_2 \oplus t_2 \tilde{<} t$ since the maximal term of $\widetilde{t}$ is neither in $\widetilde{w_2}$ nor in $\widetilde{t_2}$. We deduce that the literal $P(w_2 \oplus t_2)$ is already interpreted in $\mathcal{I}$ and $\mathcal{I}(P(w_2 \oplus t_2)) = 1$ otherwise the clause C_1 would be falsified by $\neg P(w_2) \vee \neg P(t_2) \vee P(w_2 \oplus t_2))$. Let us consider $C_1\sigma_1' = \neg P(w_2 \oplus t_2) \vee \neg P(w_1) \vee P(t)$: we are back to the previous case where C_1 and C_2 are both in S_0. We conclude applying lemma 14.

Lemma 16 (Case $C_1 \notin S_0$, $C_2 \in S_0$). *If $C_1 \notin S_0$, $C_2 \in S_0$, then $\mathcal{I}$ falsifies already a ground instance of some clause in $S^* \cup S_0$.*

Proof. $C_2\sigma_2 = \neg P(x)\sigma_2 \vee \neg P(y)\sigma_2 \vee P(x \oplus y)\sigma_2$ and $x\sigma_2 = v$. We may assume that $(x \oplus y)\sigma_2 \downarrow \tilde{<} y\sigma_2$ since when it is not the case, we may replace C_2 by $C_2' = \neg P(x') \vee \neg P(y') \vee P(x' \oplus y')$ and σ_2 by σ_2' such that $x'\sigma_2' = x\sigma_2 = v$, $y'\sigma_2' = (x \oplus y)\sigma_2$ and $(x' \oplus y')\sigma_2' = y\sigma_2$. With this transformation, this case is similar to the previous one (applying the rule **Extension 2**).

Lemma 17 (Case $C_1, C_2 \notin S_0$). *If $C_1, C_2 \notin S_0$, then $\mathcal{I}$ falsifies already a ground instance of some clause of $S^* \cup S_0$.*

Proof. The binary resolution allows us to conclude.

E Secrecy of our protocol

Proposition 3. *The set of clauses representing our protocol together with the security property $\mathcal{C}_P \cup \{\phi_0\}$ is satisfiable.*

Proof. We split the set of constants $\mathit{\Gamma}$ into the set of (supposedly) secret data $\mathit{\Gamma}_1$ and known data $\mathit{\Gamma}_2$: $\mathit{\Gamma}_1 = \{n_{ab}^1, n_{ba}^1, n_{ab}^2, n_{ba}^2, S_{ab}, S_{ba}, K_{ab}\}$ and $\mathit{\Gamma}_2 = \mathit{\Gamma} \backslash \mathit{\Gamma}_1$. We consider a set of terms T (resp. T') such that an even (resp. odd) number of “secrets” data is xored:

$$T = \{u_1 \oplus \cdots \oplus u_n \oplus t_1 \oplus \cdots \oplus t_k \mid n \text{ is even, } u_i \in \mathit{\Gamma}_1, t_j \in \mathit{\Gamma}_2, u_i, t_j \text{ distinct}\}.$$

Then we consider the following set of clauses:

$$\begin{aligned} S^* \stackrel{\text{def}}{=} &\{I(m) \mid m \in T\} \cup \{\neg I(z \oplus m_1) \vee I(z \oplus m_2) \mid m_1 \oplus m_2 \in T\} \\ &\cup \{\neg I(m_1) \vee I(m_2) \mid m_1 \oplus m_2 \in T\} \cup \{\neg I(m) \mid m \in T'\} \end{aligned}$$

S^* contains $\mathcal{C}_P \cup \{\phi_0\}$, thus it is sufficient to prove that S^* is satisfiable (actually S^* is obtained from $\mathcal{C}_P \cup \{\phi_0\}$ by applying our deduction rules thus S^* is satisfiable iff $\mathcal{C}_P \cup \{\phi_0\}$ is satisfiable).

Let us show that S^* is already saturated by our deduction rules together with the redundancy criterion.

The **Factorization** rule can not be applied to S^*. The **Narrowing** rule clearly preserves membership to S^*. The **Explosion** rule does not modify the sum $m_1 \oplus m_2$ thus preserves membership to S^*.

Let us consider a **binary resolution** between two clauses C_1 and C_2 of S^* with variables. $C_1 = \neg I(z \oplus m_1) \vee I(z \oplus m_2)$ and $C_2 = I(z' \oplus m_3) \vee \neg I(z' \oplus m_4)$, with

$m_1 \oplus m_2, m_3 \oplus m_4 \downarrow \in T$. Let $\theta \in mgu(z \oplus m_1, z' \oplus m_3)$. $z\theta = z'' \oplus m_5$, $z'\theta = z'' \oplus m_6$, such that $(m_5 \oplus m_6) \downarrow = (m_1 \oplus m_3) \downarrow$. The resulting clause is $C = \neg I(z'' \oplus m_6 \oplus m_4) \vee I(z'' \oplus m_5 \oplus m_2)$. Since $(m_6 \oplus m_4 \oplus m_5 \oplus m_2) \downarrow = (m_1 \oplus m_3 \oplus m_2 \oplus m_4) \downarrow$ is in T, it follows that C in S^*. The case where one of the clause is ground is similar.

Let us consider the rule **Extension 1** (the rule **Extension 2** is similar). Let $C_1 = \neg I(z \oplus m_1) \vee I(z \oplus m_2)$ and $C_2 = I(z' \oplus m_3) \vee \neg I(z' \oplus m_4)$, with $(m_1 \oplus m_2) \downarrow, (m_3 \oplus m_4) \downarrow \in T$. We have $t = z \oplus m_1 = z \oplus m_1' \oplus m_1''$ and $t_1 = z \oplus m_1'$ since $V(t) = V(t_1)$. $u_1 = z' \oplus m_3'$ and $u_2 = m_3''$ such that $m_3' \oplus m_3'' = m_3$ or $u_2 = z' \oplus m_3'$ and $u_1 = m_3''$. We only consider the first case since the second one is similar. Let $\theta \in mgu(z \oplus m_1', z' \oplus m_3')$. $z\theta = z'' \oplus m_5$, $z'\theta = z'' \oplus m_6$, such that $(m_5 \oplus m_6) \downarrow = (m_1' \oplus m_3') \downarrow$. The resulting clause is

$$C = I(z'' \oplus m_5 \oplus m_2) \vee \neg I(z'' \oplus m_6 \oplus m_4) \vee \neg I(m_1'' \oplus m_3'').$$

There are two cases:

- either $m_1'' \oplus m_3'' \in T$, in which case $(m_1' \oplus m_3' \oplus m_2 \oplus m_4) \downarrow \in T$ since $(m_1 \oplus m_2 \oplus m_3 \oplus m_4) \downarrow \in T$. In this case, $m_5 \oplus m_2 \oplus m_6 \oplus m_4 \in T$ since $m_5 \oplus m_6 = m_1' \oplus m_3'$.It follows that $I(x \oplus m_5 \oplus m_2) \vee I(x \oplus m_6 \oplus m_4) \in S^*$, a clause which subsumes C.
- or else $m_1'' \oplus m_3'' \notin T$, in which case $m_1'' \oplus m_3'' \in T'$ and therefore $\neg I(m_1'' \oplus m_3'') \in S^*$, a clause which, again, subsumes C.

If C_2 is a ground clause, we get exactly the same inference as above, except that z'' is absent. The same reasoning applies.

S^* is saturated by our inference rules (see appendix E). Applying theorem 2, since $\bot \notin S^*$, it follows that $\mathcal{C}_P \cup \{\phi_0\}$ is satisfiable. □

We conclude that S^* is already saturated by our deduction rules together with the redundancy criterion.

Since the abstraction is an upper approximation, the above proposition shows that the protocol is secure.