Tractable inference systems: an extension with a deducibility predicate

Hubert Comon-Lundh[1], Véronique Cortier[2], Guillaume Scerri[1 2]

[1] LSV, CNRS & ENS Cachan, France
[2] LORIA, CNRS, France

Abstract. The main contribution of the paper is a PTIME decision procedure for the satisfiability problem in a class of first-order Horn clauses. Our result is an extension of the tractable classes of Horn clauses of Basin & Ganzinger in several respects. For instance, our clauses may contain atomic formulas $S \vdash t$ where $\vdash$ is a predicate symbol and S is a finite set of terms instead of a term. $\vdash$ is used to represent any possible computation of an attacker, given a set of messages S. The class of clauses that we consider encompasses the clauses designed by Bana & Comon-Lundh for security proofs of protocols in a computational model. Because of the (variadic) $\vdash$ predicate symbol, we cannot use ordered resolution strategies only, as in Basin & Ganzinger: given $S \vdash t$, we must avoid computing $S' \vdash t$ for all subsets S' of S. Instead, we design PTIME entailment procedures for increasingly expressive fragments, such procedures being used as oracles for the next fragment.
Finally, we obtain a PTIME procedure for arbitrary ground clauses and saturated Horn clauses (as in Basin & Ganzinger), together with a particular class of (non saturated) Horn clauses with the $\vdash$ predicate and constraints (which are necessary to cover the application).

1 Introduction

1.1 The application context

The design of automated security proofs is a topic extensively studied for over 20 years. One problem that was raised about 12 years ago is the validity (or the scope) of such proofs. More specifically, for most of the automatic security proofs messages are abstracted by terms and the attackers capabilities are restricted to a specific set of operations. In contrast, modern cryptography typically considers attackers that can perform any computation that does not require too much time (say, in probabilistic polynomial time). This includes of course some computations that are not explicitly specified. This issue has been first addressed by M. Abadi and P. Rogaway [1], followed by many authors. The idea is to prove that the symbolic formal model is *sound* with respect to the more concrete computational model: if there is no attack in the symbolic model, then there is no attack in the computational model, except with negligible probability. There are several such soundness proofs, for various primitives and in various contexts (see e.g. [11, 2, 9] to cite only a few). However, all these results require heavy proofs and assume strong hypotheses, some of which are not quite realistic. Typical examples of unrealistic assumptions include: a key cycle is never created, or the attacker does use his own keys.

These difficulties lead to try to prove the security protocols directly in the computational model. For instance CryptoVerif [7] or Easycrypt [5] are designed in this spirit. The proofs have however to account for probability distributions computations, attacker's time computation, and are relatively difficult, often requiring user interactions. We study here an alternative approach presented in [4] which consists in specifying formally what the attacker *cannot do.* Each axiom in such a specification can be a consequence of an assumption on the primitives, which yields the soundness of the model by construction. The drawback is however the proof automation in this model: there was no evidence that this is possible in a reasonably efficient way. This is the problem that we want to address in this paper.

In the model of [4], transitions of the system are possible, as soon as they do not contradict the axioms. Hence, an attack consists in a sequence of attacker's actions, that is consistent with the axioms and the negation of the security property. Conversely, if all (symbolic) transition sequences yield a formula, which is inconsistent with the axioms and the negation of the security property, then the protocol is secure, for any attacker, in any model that satisfies the axioms. The clauses make use of a *deducibility predicate* $\vdash$, whose interpretation is not fixed: it stands for any attacker's computation. In other words, $S \vdash h$ states that the attacker must be able to compute h from his knowledge at this stage.

In summary, checking for cryptographic security amounts to checking the satisfiability of a finite set of ground formulas Φ together with axioms A (which are Horn clauses) and the negation of the security property π (a ground fact). Since, in practice, this satisfiability check has to be performed for any interleaving of (symbolic) actions, it must be efficiently performed. Fortunately, the formulas are not arbitrary first-order formulas. We introduce them informally below.

- Φ contains only literals (positive or negative). We actually prove that satisfiability is in PTIME as soon as Φ only contains (ground) Horn clauses.
- A could be arbitrary, in principle, provided that it is consistent with π. In practice, we may assume that $A \cup \{\pi\}$ is a finite set of (possibly constrained) Horn clauses with equality (see [3] for a complete example). A typical example of an axiom (a consequence of IND-CCA, see [4]) is the *secrecy axiom*

$$\forall X, x, y. \quad \big[X; \mathsf{enc}(x, pk) \vdash n(y) \quad \rightarrow \quad X \vdash n(y)\big] \quad || \quad sk \notin X$$

  The expression $n(y)$ represents a function that returns a random number. The formula states that the encryption of x does not help in deducing the nonce $n(y)$, unless the decryption key sk appears as a plain text of some term in X.

The problem that we consider in this paper is then the following one: when is such a satisfiability check tractable?

1.2 Difficulties

Following the approach of D. Mc Allester [10], D. Basin and H. Ganzinger [6] show that, if a set of Horn clauses is saturated, with respect to a well suited ordering and a well suited notion of redundancy, then the associated inference system is tractable. The main restriction in this paper is on the ordering with respect to which the clauses have to be saturated: given a ground term t, there should be only polynomially many terms smaller than t. (The subterm ordering, is an example. The term embedding does not satisfy this property).

However, the Horn clauses derived from security assumptions are beyond the scope of these results for several reasons that we describe below.

- The deducibility predicate $\vdash$ can be seen as a variadic predicate symbol, whose arguments (except the last one) are unordered. This is a problem, since Basin and Ganzinger method yields an NP decision procedure with such a predicate: even if A is saturated (modulo the set axioms for the left part of the $\vdash$ predicate), when we use A to reduce a ground atom $S \vdash t$, potentially all subsets of S will be considered (see Section 3 for an example).
- Axioms (i.e. Horn clauses) are constrained. A priori, this is not an obstacle to the Basin and Ganzinger procedure, as the constraints can be checked on each superposition between an axiom and a ground clause. However, the very notion of saturation of a set of constrained clauses is an issue (as reported for instance in [12] for basic strategies or [8] for order constraints). In short: we cannot assume our set of axioms to be saturated.
- Clauses contain an equality predicate. This is not too tricky, since we may assume that A does not contain any equality. Hence equalities appear only as ground literals. We can then easily extend Basin and Ganzinger algorithm to clauses modulo a ground equational theory.

1.3 Overview of the results and proofs

Including a variadic predicate. We consider sets of ground Horn clauses with equality, whose atomic formulas may (also) be $S \vdash t$ where S is a finite set of (ground) terms and t is a ground term, together with a saturated set of clauses $\mathcal{A}$ with no deducibility predicate and the following set of clauses A_0:

$$A_0 = \begin{cases} X \vdash x \to X; y \vdash x & \text{weakening} \\ X \vdash x, \quad Y; x \vdash y \to X; Y \vdash y & \text{transitivity} \\ \to x \vdash x & \\ X_1 \vdash x_1, \ldots, X_n \vdash x_n \to X_1; \ldots; X_n \vdash f(x_1, \ldots, x_n) & f \text{ function symbol} \end{cases}$$

Note that the left argument of $\vdash$ is a set. We write $X; x$ for $X \cup \{x\}$ and $X; Y$ for $X \cup Y$ and we compute modulo the set properties.

We prove first that satisfiability of such a set of clauses is in PTIME, therefore extending Basin and Ganzinger result, on the one hand with equalities (this is not the difficult part) and on the other hand with the deducibility predicate.

The main idea then is to use another layer of ground Horn clauses entailment problem: given $S_1 \vdash t_1, \ldots, S_n \vdash t_n, S \vdash t$, whether $S_1 \vdash t_1, \ldots, S_n \vdash t_n$ entails $S \vdash t$ can be solved in PTIME. This is done by transforming literals $S \vdash t$ into clauses $S \to t$. Since the resulting clauses do not contain $\vdash$ anymore, this can be used as an oracle in a (modified) ground Horn clauses entailment problem.

Adding axioms on the deducibility predicate. The previous result is not sufficient for our purpose as, for instance, simple axioms such as secrecy (provided in Section 1.1) cannot be expressed in the considered fragment.

We therefore extend the previous results, adding formulas of the form

$$S \vdash x, \quad S; u(x) \vdash t(y) \quad \to \quad S \vdash t(y)$$

$$S; u(x) \vdash v(y) \quad \to \quad S \vdash v(y)$$

These formulas are relevant for our application. Indeed, the secrecy axiom described in Section 1.1 is an axiom of the second form. The axioms of the first form are useful to express e.g. non malleability of encryption:

$$\forall X, x, y. \quad X \vdash x \quad X; \mathsf{dec}(x, k) \vdash n(y) \quad \to \quad X \vdash n(y) \qquad \| \quad P(x)$$

The decryption of a deducible message x does not help to learn a nonce $n(y)$, provided that x does not appear as subterm of X, which can be encoded in a predicate P.

We show again in this case that the satisfiability is in PTIME. The first idea consists in seeing these clauses as new inference rules. For instance the first above axiom can be seen as a generalized cut (it is a cut when $u(x) = x$). As before, we first consider the entailment problem for deduction atomic formulas, which in turn can be seen as an entailment problem for Horn clauses. This can also be easily reduced to the problem of deducing the empty clause.

We design a strategy, which is complete for this extended deduction system and for which the proof search is in PTIME. Let us explain how it works. With the usual cut rule (and not the extended one above), whether the empty clause can be derived, can be decided in PTIME using a unit strategy. This is not the case with an extended cut rule. However, introducing some new rules and additional syntactic constructions, we design a proof system, whose expressive power is the same as the original proof system, and for which the unit strategy is complete, yielding a PTIME decision procedure. In other words, our strategy, that cannot be explained as a local strategy of application, can be reduced to a unit strategy, thanks to some memorization.

Adding constraints. Our application case requires to consider constraints, typically expressing that some term does not occur in the left side of a deduction relation. Such constraints have good stability properties: if they are satisfied by two sets of literals, then they are satisfied by their union and, if a constraint is satisfied by a set of literals S, then it is satisfied by any subset of S. Our main restriction is however that there are only a fixed set of possible constraints. We show again that the satisfiability is in PTIME.

We cannot simply use the previous strategy, checking that constraints are satisfied whenever we need to apply them. The extended deduction system of the previous section is proved to be complete by a proof transformation that may not preserve constraint satisfaction. We therefore refine the strategy, memorizing additional information in the formulas: on the one hand, we store the constraints that are necessarily satisfied by all instances of the clause (this is inherited in the deduction rules) and, on the other hand, the constraints that have to be satisfied in the remainder of the proofs. Using this new syntax and inference rules, we show that they do not increase the expressiveness and yet that the unit strategy is refutation complete for these new rules. This shows the PTIME membership.

In the next step, we show that the entailment problem is decidable in PTIME in this new syntax. We need however to memorize a third component, which depends on the instance of the entailment problem.

Final result. From the previous paragraphs, we can build a PTIME entailment algorithm which, given $S_1 \vdash t_1 \ldots S_n \vdash t_n, S \vdash t$ and clauses

$$A_1 = \begin{cases} S \vdash x, \quad S; u_i(x) \vdash t(y) \quad \rightarrow \quad S \vdash t(y) \quad \| \quad \Gamma_i \\ S; s_j(x) \vdash v(y) \quad \rightarrow \quad S \vdash v(y) \quad \| \quad \Delta_j \end{cases}$$

where Γ_i, Δ_j are finite sets of constraints, decides in PTIME whether $S_1 \vdash t_1, \ldots, S_n \vdash t_n, A_1, A_0, \mathcal{A} \models S \vdash t$.

This algorithm can be used as an oracle in a variant of the Basin and Ganzinger algorithm, to decide the satisfiability of a set of clauses including formulas extending A_0, A_1 together with ground clauses with equality. Altogether, we obtain a PTIME procedure for arbitrary ground clauses and saturated Horn clauses (as in Basin & Ganzinger), together with the aforementioned clauses. This is exactly what we needed for our application, that is checking satisfiability of clauses corresponding to the computational security of a protocol.

Beyond our tractability results, we hope that our techniques and ideas of memorization can be reused in other contexts for the design of efficient strategies.

2 Formal setting

Let $\mathcal{F}$ be a finite set of function symbols (together with their arity) and $\mathcal{P}$ be a finite set of predicate symbols together with their arity. $T(\mathcal{F})$ is the set of ground terms built on $\mathcal{F}$ (which is assumed to contain at least one constant) and $T(\mathcal{F}, \mathcal{X})$ is the set of terms built on $\mathcal{F}$ and a set of variable symbols $\mathcal{X}$. We also use set variables (written using upper case letters $X, Y, Z, \ldots$) ranging in a set $\mathcal{SX}$ and a function symbol, denoted by a semicolon, for set union. *Extended terms* $ET(\mathcal{F}, \mathcal{X}, \mathcal{SX})$ are expressions $s_1; \ldots; s_n$ where $s_i \in T(\mathcal{F}, \mathcal{X}) \cup \mathcal{SX}$. As a shortcut, when $n = 0$ in the previous definition we denote the extended term as $\emptyset$. A *basic ordering* is an ordering on terms, which is : 1. Compatible with substitutions and 2. such that, for every ground term t, the number of terms smaller than t is polynomial in the size of t. (An example of such an ordering is the subterm ordering).

Atomic formulas are of the following forms:

- $P(t_1, \ldots, t_n)$ where $P \in \mathcal{P}$ and $t_1, \ldots, t_n \in T(\mathcal{F}, \mathcal{X})$
- $t_1 = t_2$ where $t_1, t_2 \in T(\mathcal{F}, \mathcal{X})$
- $S \vdash t$ where $t \in T(\mathcal{F}, \mathcal{X})$ and $S \in ET(\mathcal{F}, \mathcal{X}, \mathcal{SX})$.

We consider clauses that are built on these atomic formulas. The axioms for the set theory ACIN (associativity, commutativity, idempotence and neutral element $\emptyset$) are implicitly assumed without mention on the left side of the $\vdash$. As usual, Horn clauses are clauses with at most one positive literal.

Given an extended term S and a substitution σ, mapping variables of $\mathcal{SX}$ to finite subsets of $T(\mathcal{F})$ and variables of $\mathcal{X}$ to terms in $T(\mathcal{F})$, $S\sigma$ is defined by $\emptyset\sigma = \emptyset$, $(s; S)\sigma = \{s\sigma\} \cup S\sigma$ if $s \in T(\mathcal{F}, \mathcal{X})$, and $(X; S)\sigma = X\sigma \cup S\sigma$ if $X \in \mathcal{SX}$.

3 Tractability of deducibility axioms

We first consider the consistency problem of a very specific case: let $\mathcal{C}$ be a set of ground clauses built on the deducibility predicate only. Is $\mathcal{C} \cup \{\to X; x \vdash x, \quad X \vdash x \to X; y \vdash x, \quad X \vdash x,\ X; x \vdash y \to X \vdash y\}$ consistent? (We call respectively *r(eflexivity)*, *w(eakening)* and *t(ransitivity)* the three last clauses).

Consider for instance a ground clause $a_1, \ldots, a_n \vdash a \to\perp$. If we simply use a unit resolution strategy (which is refutation complete for Horn clauses), this single clause, together with the weakening clause, may generate all unit clauses $S \vdash a \to\perp$ where $S \subseteq \{a_1, \ldots, a_n\}$. This should be avoided since we seek for a polynomial time algorithm. Similar problems occur with transitivity, if we try to use binary resolution with a simple strategy. Here is a more concrete example.

Example 1. Let $\mathcal{C} = \{a_1; a_2; a_3 \vdash a_0 \to\perp, \quad \to a_1; a_4 \vdash a_0, \quad \to a_2 \vdash a_4\}$. $\mathcal{C} \cup \{w, t\}$ is provably unsatisfiable using binary resolution modulo ACIN only.

$$\frac{\dfrac{\to a_1; a_4 \vdash a_0 \quad X_1 \vdash x_1 \to X_1; y_1 \vdash x_1}{\to a_1; a_4; y_1 \vdash a_0} \qquad X_2 \vdash x_2,\ \ X_2; x_2 \vdash y_2 \to X_2 \vdash y_2}{a_1; y_1 \vdash a_4 \to a_1; y_1 \vdash a_0}$$

with unifiers $X_1 = a_1; a_4$, $X_2 = a_1; y_1$, $x_1 = a_0$, $x_2 = a_4$ and $y_2 = a_0$

$$\frac{a_1; y_1 \vdash a_4 \to a_1; y_1 \vdash a_0 \qquad \dfrac{\to a_2 \vdash a_4 \quad X_3 \vdash x_3 \to X; y_3 \vdash x_3}{\to a_2; y_3 \vdash a_4}}{\to a_1; a_2 \vdash a_0}$$

with unifiers $X_3 = a_2$, $y_1 = a_2$ and $y_3 = a_1$

and $$\frac{\dfrac{\to a_1; a_2 \vdash a_0 \quad X_4 \vdash x_4 \to X_4; y_4 \vdash x_4}{\to a_1; a_2; y_4 \vdash a_0} \qquad a_1; a_2; a_3 \vdash a_0 \to\perp}{\perp}$$

with unifiers $X_4 = a_1; a_2$, $x_4 = a_0$ and $y_4 = a_3$.

This derivation introduces the clause $\rightarrow a_1; a_2 \vdash a_0$, where $a_1; a_2$ is a new set (i.e. it does not appear in the initial sets). This is actually unavoidable: any derivation of the empty clause requires as an intermediate step the derivation of either $\rightarrow a_1; a_2 \vdash a_0$ or $a_1; a_4; a_3 \vdash a_0 \rightarrow\perp$. Both of them involve sets that are not in the initial class.

However if we move from the object level to the meta-level, viewing weakening and transitivity as inference rules and deducibility atoms as clauses, we can at least solve this very particular case. More precisely, consider the inference system:

$$\frac{}{X; x \vdash x}\,R \qquad \frac{X \vdash x}{X; y \vdash x}\,W \qquad \frac{X \vdash x \quad X; x \vdash y}{X \vdash y}\,T$$

where X is a logical variable ranging over extended terms and x, y are logical variables ranging over terms.
Let $\Vdash_{R,W,T}$ be the derivability relation associated with these two inference rules.

Lemma 1. *Given ground atomic formulas $S_1 \vdash t_1, \ldots, S_n \vdash t_n$ and $S \vdash t$, we can decide in linear time whether $\{S_1 \vdash t_1, \ldots, S_n \vdash t_n\} \Vdash_{R,W,T} S \vdash t$.*

Proof. We associate with each term occurring in $S_1 \cup \ldots \cup S_n \cup S \cup \{t_1, \ldots, t_n, t\}$ a proposition variable. We claim that $S_1 \vdash t_1, \ldots, S_n \vdash t_n \Vdash_{R,W,T} S \vdash t$ iff $S \rightarrow t$ is derivable from $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n$ using the propositional binary resolution, excluded middle and weakening rules only. Indeed we notice that T, R and W can be simulated by resolution and excluded middle. For W the proof rewriting is straightforward. We present the proof rewriting for T and R :

$$\frac{S \vdash t \quad S; t \vdash u}{S \vdash u}\,T \quad \Longrightarrow \quad \frac{S \rightarrow t \quad S, t \rightarrow u}{S \rightarrow u}\,Res$$

$$\frac{}{S; t \vdash t}\,R \Longrightarrow \frac{\dfrac{}{t \rightarrow t}\,Excl}{S, t \rightarrow t}\,Weak$$

Conversely the resolution, excluded middle and weakening can be simulated by R, T and W. The proof rewriting is straightforward for excluded middle and weakening, we only present it for resolution :

$$\frac{S_1 \rightarrow t \quad S_2, t \rightarrow u}{S_1, S_2 \rightarrow u}\,Res \quad \Longrightarrow \quad \frac{\dfrac{S_1 \vdash t}{S_1; S_2 \vdash t}\,W \qquad \dfrac{S_2; t \vdash u}{S_1; S_2; t \vdash u}\,W}{S_1; S_2 \vdash u}\,T$$

Then derivability of $S \rightarrow t$ is equivalent to unsatisfiability of $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n, S, \neg t$ (where S_i is a shortcut for the conjunction of propositional variables corresponding to terms occurring in S_i), which can be decided in linear time: it is a HornSat problem.

Now, the trick of viewing the clauses w, t as new inference rules allows to decide our problem in PTIME. We write $\Vdash_{Res_u+R+W+T}$ for the derivability with inference rules R, W, T and unit resolution.

Lemma 2. *Given a set of ground Horn clauses (built on $\vdash$) $\mathcal{C}$, the satisfiability of $\mathcal{C} \cup \{r, w, t\}$ is decidable in cubic time.*

Proof. We show first that $\mathcal{C} \cup \{r, w, t\}$ is unsatisfiable iff the empty clause can be derived from $\mathcal{C}$, using unit resolution $R + W + T$. If we can derive the empty clause in this system, then we can derive the empty clause from $\mathcal{C} \cup \{r, w, t\}$ by resolution, thanks to simple proof rewriting rules :

$$\frac{}{S;t\vdash t}\,R \qquad\Longrightarrow\qquad S;t\vdash t \text{ (instance of } r\text{)}$$

$$\frac{\overset{\pi_1}{S\vdash t}}{S;u\vdash t}\,W \qquad\Longrightarrow\qquad \frac{\overset{\pi_1}{S\vdash t} \qquad X\vdash x \;\to\; X;y\vdash x}{S;u\vdash t}\,Res$$

$$\frac{\overset{\pi_1}{S\vdash t} \quad \overset{\pi_2}{S;t\vdash u}}{S\vdash u}\,T \qquad\Longrightarrow\qquad \frac{\dfrac{\overset{\pi_1}{S\vdash t} \quad X;x\vdash y,\ X\vdash x \;\to\; X\vdash y}{S;t\vdash y \to S\vdash y}\,Res \qquad \overset{\pi_2}{S;t\vdash u}}{S\vdash u}\,Res$$

Conversely, if we cannot derive the empty clause from $\mathcal{C}$ using unit resolution $R + W + T$, then let $\mathcal{M} = \{S \vdash u \mid \mathcal{C} \Vdash_{Res_u+R+W+T} S \vdash u\}$. We claim that $\mathcal{M}$ is a model of $\mathcal{C} \cup \{r, w, t\}$: As $\mathcal{M}$ is closed by R, W, T, it is a model of $\{r, w, t\}$ and, if $B_1, \ldots, B_n \to H \in \mathcal{C}$, then either $B_i \notin \mathcal{M}$ for some i or else, by construction, for every i, $\mathcal{C} \Vdash_{Res_u+R+W+T} B_i$, hence, by unit resolution, $\mathcal{C} \Vdash_{Res_u+R+W+T} H$. In all cases, $\mathcal{M} \models B_1, \ldots, B_n \to H$.

It only remains to prove that whether $\mathcal{C} \Vdash_{Res_u+R+W+T} \bot$ or not can be decided in cubic time. Let $\mathcal{B}$ be the set of atomic formulas occurring in $\mathcal{C}$. Let $\mathcal{M}$ be the least fixed point of

$$f(X) = \{S \vdash u \in \mathcal{B} \mid \mathcal{C} \cup X \Vdash_{Res_u} S \vdash u \text{ or } \mathcal{C} \cup X \Vdash_{R+W+T} S \vdash u\}$$

Since f is monotone, there is a least fixed point, which is contained in $\mathcal{B}$. Computing $\mathcal{M}$ can be performed in quadratic time, as there are at most $|\mathcal{B}|$ iterations and each step requires at most a linear time, thanks to the Lemma 1.

If the empty clause can be derived from $\mathcal{M}, \mathcal{C}$ using unit resolution, then $\mathcal{C} \Vdash_{Res_u+R+W+T} \bot$. Let us show the converse implication. For this, we prove, by induction on the proof size that, for every atomic formula $S \vdash t \in \mathcal{B}$, $\mathcal{C} \Vdash_{Res_u+R+W+T} S \vdash t$ implies $S \vdash t \in \mathcal{M}$.

If the last rule of the proof is a unit resolution, then the proof can be written:

$$\dfrac{\begin{array}{c}\pi_1\\ S_1 \vdash t_1\end{array} \qquad \dfrac{\begin{array}{c}\pi_2\\ S_2 \vdash t_2\end{array} \qquad \dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\pi_n\\ S_n \vdash t_n\end{array} \quad \dfrac{}{S_1 \vdash t_1, \dots, S_n \vdash t_n \to S \vdash t}\,{\scriptstyle (S_1 \vdash t_1, \dots, S_n \vdash t_n \to S \vdash t) \in \mathcal{C}}}{S_1 \vdash t_1, \dots, S_{n-1} \vdash t_{n-1} \to S \vdash t}\\ \vdots\end{array}}{S_1 \vdash t_1, S_2 \vdash t_2 \to S \vdash t}}{S_1 \vdash t_1 \to S \vdash t}}{S \vdash t}$$

$S_1 \vdash t_1, \dots, S_n \vdash t_n \in \mathcal{B}$ and, by induction hypothesis, $S_1 \vdash t_1, \dots, S_n \vdash t_n \in \mathcal{M}$. It follows that $\mathcal{M}, \mathcal{C} \Vdash_{Res_u} S \vdash t$, hence $S \vdash t \in f(\mathcal{M}) = \mathcal{M}$.

If the last rule of the proof is W or T, then there are atomic formulas $S_1 \vdash t_1, \dots, S_n \vdash t_n$ such that $S_1 \vdash t_1, \dots, S_n \vdash t_n \Vdash_{R+W+T} S \vdash t$ and, for every i, either $S_i \vdash t_i \in \mathcal{C}$ or the last rule in the proof of $S_i \vdash t_i$ is a resolution step and, as noticed previously all, $S_i \vdash t_i$ are in $\mathcal{B}$. In all cases $S_i \vdash t_i \in \mathcal{B}$ and, by induction hypothesis, $S_i \vdash t_i \in \mathcal{M}$. By definition of the function f, $S \vdash t \in f(\mathcal{M}) = \mathcal{M}$.

If $\mathcal{C} \Vdash_{Res_u+R+W+T} \perp$, then there is a negative clause $S_1 \vdash t_1, \dots, S_n \vdash t_n \to \perp$ in $\mathcal{C}$ such that, for every i, $\mathcal{C} \Vdash_{Res_u+R+T+W} S_i \vdash t_i$, hence $S_i \vdash t_i \in \mathcal{M}$ as we just saw. Then $\perp$ can be deduced from $\mathcal{C}, \mathcal{M}$ using unit resolution (which can be decided in linear time again).

Example 2. Applying Lemma 2 to Example 1, checking the satisfiability of $\mathcal{C} \cup \{r, w, t\}$ simply amounts into checking whether $\{a_1; a_4 \to a_0, \quad a_2 \to a_4\}$ (does not) entail $a_1; a_2; a_3 \to a_0$.

3.1 Adding equality

Now, we assume that atomic formulas in $\mathcal{C}$ may contain equalities on terms (not extended terms). The equality axioms (the equality is a congruence) are implicit in what follows.

Lemma 3. *Given a set of ground Horn clauses (built on $\vdash$ and $=$) $\mathcal{C}$, the satisfiability of $\mathcal{C} \cup \{r, w, t\}$ is decidable in polynomial time.*

Proof sketch: First, we extend the Lemma 1. Given a finite set of equations E, the transitivity rule is extended to

$$\dfrac{x =_E z \quad X \vdash x \quad X; z \vdash y}{X \vdash y}\, T(E)$$

Given $S_1 \vdash t_1 \dots, S_n \vdash t_n, S \vdash t$ and a finite set of ground equations E, we can decide in polynomial time whether $S_1 \vdash t_1, \dots, S_n \vdash t_n \Vdash_{R,W,T(E)} S \vdash t$. We only have to check, for every pair of terms u, v in $S_1, t_1, \dots, S_n, t_n, S, t$, whether $u =_E v$. This can be completed in polynomial time, for instance using a quadratic time congruence closure algorithm. We may then choose one representative for each congruence class and use the same proof as in the Lemma 1 on the representatives.

Then, as in the lemma 2, we consider the set $\mathcal{B}_{\vdash}$ of atomic formulas $S \vdash t$ occurring in $\mathcal{C}$ and $\mathcal{B}_{=}$ the set of equations occurring as atomic formulas in $\mathcal{C}$. We consider the monotone function

$$\begin{aligned} f(X,E) = (\ & \{S \vdash t \in \mathcal{B}_{\vdash} \mid \mathcal{C} \cup X \Vdash_{Res_u(E)} S \vdash t \text{ or } \mathcal{C} \cup X \Vdash_{R+W+T(E)} S \vdash t\}, \\ & \{s = t \in \mathcal{B}_{=} \mid \mathcal{C} \cup X \Vdash_{Res_u(E)} s = t\} \qquad\qquad\qquad\qquad\qquad) \end{aligned}$$

where $\Vdash_{Res_u(E)}$ is the unit resolution on representatives of the clauses w.r.t. E.

The least fixed point of f can be computed in polynomial time, as each iteration is polynomial and there is a polynomial number of iterations. $\mathcal{C}\cup\{r, w, t\}$ is satisfiable iff the empty clause can not be derived by unit resolution from this least fixed.

3.2 Adding a function axiom

We extend now the clauses specifying $\vdash$ with the clauses (denoted by $f(\mathcal{F})$ later): $X \vdash x_1, \quad \cdots \quad X \vdash x_n \quad \rightarrow \quad X \vdash g(x_1, \ldots, x_n)$, for every function symbol g in a set of function symbols $\mathcal{F}$ (which is later omitted).

Lemma 4. *Given a set of ground Horn clauses (built on $\vdash$ and $=$) $\mathcal{C}$, the satisfiability of $\mathcal{C} \cup \{r, w, t\} \cup f(\mathcal{F})\}$ is decidable in polynomial time.*

Proof sketch: Again, adding an inference F_g for each of the new clauses, we first show that deciding $S_1 \vdash t_1, \ldots, S_n \vdash t_n \Vdash_{R+W+T(E)+\{F_g, g\in\mathcal{F}\}} S \vdash t$ is in PTIME. We use a proof similar to the Lemma 3, with an additional observation: given a finite set E of ground equations and ground terms $t_1, \ldots, t_n, t$, we can decide in PTIME whether there is a context C (built using function symbols in $\mathcal{F}$) such that $C[t_1, \ldots, t_n] =_E t$. To prove this we may for instance compute a tree automaton $\mathcal{A}_t$ that recognizes the equivalence class of t and decide the emptiness of the intersection of $L(\mathcal{A}_t)$ with the set of terms $C[t_1, \ldots, t_n]$. All these steps can be performed in a total time, which is polynomial in the size of E, $t_1, \ldots, t_n, t$.

Example 3. $b \vdash c$, $\vdash a \Vdash_{R+W+T(g(g(a))=b)+F_g} \vdash c$ since there is a context C (with $C[\_] = g(g(\_))$) such that $C[a] = b$.

4 More clauses using the deducibility predicate

We now enrich the class of clauses involving the deducibility predicate. Given a term p (later called the *pattern*), we consider a finite set of clauses of the following forms:

$c_s(u)\ : X; u \vdash p \rightarrow X \vdash p$ where u is a term that does not share variables with p

$c_c(w)$: $X \vdash y, X; w \vdash p \rightarrow X \vdash p$ where w is a term that does not share variables with p, and y is a variable of w.

Example 4. The secrecy axiom described in introduction

$$X; \mathsf{enc}(x, pk) \vdash n(y) \quad \rightarrow \quad X \vdash n(y)$$

is an instance of the first class of clauses above, with $p = n(y)$ and $u = \mathsf{enc}(x, pk)$. The condition $sk \notin X$ requires constraints, that are considered in Section 5.

As explained in the previous section, we may turn the additional clauses into new inference rules, using $\leq_E$, the matching modulo E (a term t satisfies $u \leq_E t$ if there is a substitution σ such that $t =_E u\sigma$).

$$\frac{u \leq_E x \quad X; x \vdash p}{X \vdash p}\,\mathsf{Str}_u \qquad\qquad \frac{(y, w) \leq_E (x, z) \quad X \vdash x \quad X; z \vdash p}{X \vdash p}\,\mathsf{Cut}_w$$

Let $\mathcal{I}$ be the inference system defined by a finite collection of rules $\mathsf{Str}_u, \mathsf{Cut}_w$, the rules $R, W, T(E)$ for a finite set of ground equations E and the rules F_g for a set of function symbols g.

We are going to prove that, again, $\mathcal{I}$ can be decided in polynomial time. However, we cannot use the same proof as in the previous section. $S_1 \vdash t_1, \ldots, S_n \vdash t_n \Vdash_{\mathcal{I}} S \vdash t$ can no longer be reduced to a problem $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_1, S \Vdash_{Res_u} t$ (modulo a PTIME oracle).

Example 5. Assume E is empty and we have a single rule $\mathsf{Cut}_{f(x,k)}$ for the pattern $p = n$. $f(a, k) \vdash f(b, k),\ f(b, k) \vdash n \Vdash_{\mathcal{I}} a \vdash n$:

$$\dfrac{\dfrac{}{a \vdash a}\,R \qquad \dfrac{\dfrac{f(a,k) \vdash f(b,k)}{a; f(a,k) \vdash f(b,k)}\,W \qquad \dfrac{f(b,k) \vdash n}{a; f(a,k); f(b,k) \vdash n}\,W}{a; f(a,k) \vdash n}\,T}{a \vdash n}\,\mathsf{Cut}_{f(x,k)}$$

We cannot use a unit version of T (or resolution) in this example. And moving to a general binary resolution would yield an exponential procedure.

As before, after turning the clauses into inference rules, we turn the deducibility atomic formulas into clauses. We call again $\mathcal{I}$ the resulting inference system. We have to be careful however: this is a purely syntactic transformation and the inference rules resulting from this translation are no longer correct in a classical semantics. For instance Cut_w becomes

$$\frac{A_1, \ldots, A_n \rightarrow y \qquad w, B_1, \ldots, B_m \rightarrow p}{A_1, \ldots, A_n, B_1, \ldots, B_m \rightarrow p}$$

where the premises are matched modulo a set of ground equations E.

In order to apply a simple fixed point computation, we would like to be able to transform any proof into a unit strategy proof. Since this is not possible with the current proof system (as shown by Example 5), we introduce additional inference rules that will allow such a strategy, however bookkeeping what the rest of the proof owes, in order to enable a translation back into the original proof system.

Example 6. Continuing Example 5, the unit proof of $\to n$ from the hypotheses $\to a,\ f(a,k) \to f(b,k),\ f(b,k) \to n$ will look like this:

$$\dfrac{\dfrac{\to a \qquad f(a,k) \to f(b,k)}{\to_p f(b,k)}\,\mathsf{Cut}^1_{f(x,k)} \qquad f(b,k) \to n}{\to n}\,\mathsf{Cut}^2$$

The rule Cut^1_u is a generalisation of Cut_u since the constraint of being an instance of the pattern p on the right is dropped. It bookkeeps however a duty as a mark p on the arrow. The mark on a clause $S \to_p t$ can in turn be erased only when a clause $S', t \to p$ is one of the premises. Such a mechanism allows both to use a complete unit strategy and to enable reconstructing an original proof from the extended one, as we will prove (here the annotation is erased in the last rule as the second premise is an instance of $S, f(x,k) \vdash n$).

Intuitively, the head s of a marked clause can only be used in a proof that will end up deriving an instance of the pattern.

We extend the syntax, allowing both unmarked clauses $S \to t$ and marked clauses $S \to_p t$. For simplicity, we first do not consider the set of ground equations E nor the function axioms. We write $S \to_? t$ when it does not matter whether the arrow is marked or not. We then consider the inference system $\mathcal{J}$ consisting of $T(E)$, W and the following rules (for each Cut_w there are two rules Cut^i_w and for each rule Str_u there are two rules Str^i_u):

$$\dfrac{A_1, \ldots, A_n \to_? x \qquad B_1, \ldots, B_m, w \to_? v}{A_1, \ldots, A_n, B_1, \ldots, B_m \to_p v}\,\mathsf{Cut}^1_w$$

$$\dfrac{A_1, \ldots, A_n \to_? x \qquad w, B_1, \ldots, B_m \to p}{A_1, \ldots, A_n, B_1, \ldots, B_m \to p}\,\mathsf{Cut}^2_w$$

$$\dfrac{A_1, \ldots, A_n \to_? x \qquad B_1, \ldots, B_m, x \to_? v}{A_1, \ldots, A_n, B_1, \ldots, B_m \to_? v}\,\mathsf{Cut}^1$$

in which the conclusion is marked iff one of the premises is marked.

$$\dfrac{A_1, \ldots, A_n \to_? x \qquad x, B_1, \ldots, B_m \to p}{A_1, \ldots, A_n, B_1, \ldots, B_m \to p}\,\mathsf{Cut}^2$$

$$\dfrac{A_1, \ldots, A_n, u \to_? x}{A_1, \ldots, A_n \to_p x}\,\mathsf{Str}^1_u \qquad\qquad \dfrac{A_1, \ldots, A_n, u \to_? p}{A_1, \ldots, A_n \to p}\,\mathsf{Str}^2_u$$

Note that the above system has no classical semantics.

Lemma 5. *Let $\mathcal{S}$ be a set of ground clauses, and s be a ground term. In case $E = \emptyset$ and removing the function and reflexivity axioms from $\mathcal{I}$, $\mathcal{S} \Vdash_{\mathcal{I}} \to s$ if and only if $\mathcal{S} \Vdash_{\mathcal{J}} \to s$.*

Proof sketch: For one implication we prove that W is not necessary, hence $\mathcal{I}$ can be simulated by $\mathcal{J}$. For the other implication, we rewrite a proof in $\mathcal{J}$ as follows. We consider a last rule that introduces a mark. Since the marks must eventually disappear, there is also a matching rule that removes the mark. This part of proof is then rewritten as explained on the following example:

$$\dfrac{\dfrac{S_n \to t_n \qquad \begin{array}{c} \dfrac{S_2 \to t_2 \qquad \dfrac{S_1 \to t_1 \quad S, w\sigma \to v\sigma}{S_1, S \to_p v\sigma}\,\mathsf{Cut}^1_w}{S'_2 \to_p v\sigma}\,\mathsf{Cut}^1_{w_2} \\ \vdots \end{array}}{S'_n \to_p v\sigma}\,\mathsf{Cut}^1_{w_n} \qquad S_0, t\sigma' \to p\theta}{S_0, S'_n \to p\theta}\,\mathsf{Cut}^2_t$$

rewrites to

$$\dfrac{S_n \to t_n \qquad \begin{array}{c} \dfrac{S_1 \to t_1 \qquad \dfrac{S, w\sigma \to v\sigma \quad S_0, t\sigma' \to p\theta}{S_0, S, w\sigma \to p\theta}\,\mathsf{Cut}_t}{S_0, S_1, S \to p\theta}\,\mathsf{Cut}_w \\ \vdots \end{array}}{S_0, S'_n \to p\theta}\,\mathsf{Cut}_{w_n}$$

The proof rewriting terminates and we end up with a proof in $\mathcal{I}$. See Appendix A for more details.

The *unit* strategy for $\mathcal{J}$ consists in applying the rules only when $n = 0$ for the Cut^i_w rules (i.e. when the left premise of a Cut^i_w is a unit clause).

Lemma 6. *If $\mathcal{S} \Vdash_{\mathcal{J}} \to s$ then $\to s$ is derivable from $\mathcal{S}$ in $\mathcal{J}$ using the unit strategy.*

Proof sketch: We prove it by induction on the proof size. We assume w.l.o.g. that all proofs of literals (whether marked or not) labeling a node in the proof (except the root) use a unit strategy. We consider the last step that does not comply with the unit strategy. If $A_1, \ldots, A_n \to_? s$ is its conclusion, then all atoms $A_1, \ldots, A_n$ can be proved in $\mathcal{J}$ with the unit strategy. We therefore simplify the premises accordingly, which yields an inference rule complying with the unit strategy.

Theorem 1. *If $\mathcal{S}$ is a set of ground clauses built on $\vdash$, we can decide in PTIME the satisfiability of $\mathcal{S}$, together with T, W and finitely many clauses c_s, c_c, that are built on the same pattern p.*

Proof sketch: we first observe that, thanks to the lemmas 5 and 6 (and using a fixed point computation), it is possible to decide in PTIME whether, given the ground atoms $S_1 \vdash t_1, \ldots, S_n \vdash t_n, S \vdash t$, $S_1 \vdash t_1, \ldots, S_n \vdash t_n \Vdash_{\mathcal{I}} S \vdash t$. We then conclude using an argument similar to the one given in Section 3.

4.1 Adding other predicate symbols

We now consider the case where the clauses c_s, c_n, c_c are guarded with literals built on a set of predicate symbols $\mathcal{P}$ not containing $\vdash$ and that are defined using a saturated set of Horn clauses $\mathcal{A}_0$. For instance, $c_c(w)$ is extended to clauses of the form $P_1(s_1), \ldots, P_n(s_n),\ X \vdash y,\ X; w \vdash p\ \rightarrow X \vdash p$. The variables of $s_1, \ldots, s_n$ are assumed to be a subset of the variables of w, y.

We modify the rules Cut_w^i adding as premises the literals $P_1(s_1), ..., P_n(s_n)$. Lemma 5 still holds, provided we add to $\mathcal{S}$ finitely many ground atoms on the new alphabet of predicates. To see this, we need to check that the proof transformation yields the same instances of $P_i(s_i)$. Lemma 6 is unchanged. These properties rely on the fact that guards (and their instances) do neither depend on the set variable X (nor its instances) nor on the instances of the pattern.

Theorem 1 can then be extended to this case: when computing the fixed point, the instances of applicable inference rules are known at each step and we only have to check whether the corresponding instances of the guards are consequences of $\mathcal{A}_0$ (and possibly a finite set of ground atoms), which can be performed in PTIME, thanks to [6]. As a consequence, we get:

Theorem 2. *Let $\mathcal{P}$ be a set of predicate symbols, not containing $\vdash, =$ and $\mathcal{A}_0$ be a set of Horn clauses built on $\mathcal{P}$ and which is saturated w.r.t. a basic ordering. If $\mathcal{S}$ is a set of ground clauses built on $\vdash$ (possibly with guards using $\mathcal{P}$), we can decide in PTIME the satisfiability of $\mathcal{S} \cup \mathcal{A}_0$, together with T, W and finitely many clauses c_n, c_s, c_c, that are built on the same pattern p and which may be guarded by atomic formulas that use the predicate symbols in $\mathcal{P}$.*

4.2 Adding equality

We can extend again Theorem 2 to ground equalities in the atomic formulas of $\mathcal{S}$. The procedure is the same as in Lemma 3: for a fixed E, Lemmas 5 and 6 can be extended, considering representatives modulo $=_E$. Then we only have to compute a fixed point of a function f on the atomic formulas of $\mathcal{S}$, using the PTIME oracles provided by (extensions of) Lemmas 5 and 6.

5 The general case

Finally, we extend the results of the previous section to clauses with constraints.

A *constraint* Γ is a formula interpreted as a subset of $((T(\mathcal{F}))^*)^n$ (n-uples of finite sets of ground terms) if n is the number of free variables of Γ. We write $S_1, \ldots, S_n \models \Gamma$ when $(S_1, \ldots, S_n)$ belongs to this interpretation. A *constrained clause* is a pair of a clause and a constraint, which is written $\phi \quad \| \quad \Gamma$. Given a constrained clause $\phi \quad \| \quad \Gamma$, we let $[\![\phi \| \Gamma]\!] = \{\phi\sigma \| \sigma \text{ satisfies } \Gamma\}$. A model of $\phi \quad \| \quad \Gamma$ is, by definition, a model of $[\![\phi \| \Gamma]\!]$. A constraint Γ is *monotone* if

- if $S_1, \ldots, S_n \models \Gamma$ and, for every i, $S'_i \subseteq S_i$, then $S'_1, \ldots, S'_n \models \Gamma$
- if $S_1, \ldots, S_n \models \Gamma$ and $S'_1, \ldots, S'_n \models \Gamma$, then $S_1 \cup S'_1, \ldots, S_n \cup S'_n \models \Gamma$.

We typically use constraints of the form $t \notin X$ (where $t \in T(\mathcal{F})$), satisfied by any S that does not contain t as subterm. Such constraints are monotone.

Adding a fixed set of possible constraints increases significantly the difficulty: Lemmas 5 and 6 no longer hold, as shown by the following example:

Example 7. Consider the clause $c_{f(y,k)}\ :\quad X \vdash y,\ X; f(y,k) \vdash n \ \rightarrow\ X \vdash n \ \ \| f(a,k), f(b,k), f(c,k) \notin X$. Consider the ground deducibility formulas: $\mathcal{S} = \{(f(a,k) \vdash f(b,k),\ f(b,k); f(c,k) \vdash n\}$. Does $c_{f(y,k)}$ and $\mathcal{S}$ entail $a; c \vdash n$?

Following the procedure of Section 4,

$$\dfrac{\rightarrow c \qquad \dfrac{\dfrac{\rightarrow a \quad f(a,k) \rightarrow f(b,k)}{\rightarrow_p f(b,k)}\,\mathsf{Cut}^1_{f(y,k)} \quad f(b,k); f(c,k) \rightarrow n}{f(c,k) \rightarrow n}\,\mathsf{Cut}^2}{\rightarrow n}\,\mathsf{Cut}^2_{f(y,k)}$$

in which each $\mathsf{Cut}^i_{f(y,k)}$ satisfies the constraint that $f(a,k), f(b,k), f(c,k)$ do not appear in the context: the instance of X is empty in each case. The procedure would then incorrectly answers "yes" to the entailment question.

Indeed, the proof rewriting of Lemma 5 yields the following (invalid) proof, in which the constraints are *not* satisfied in the first application of $\mathsf{Cut}_{f(x,k)}$, since the corresponding instance of X is the one element set $f(c,k)$:

$$\dfrac{\rightarrow c \qquad \dfrac{\rightarrow a \qquad \dfrac{f(a,k) \rightarrow f(b,k) \quad f(b,k); f(c,k) \rightarrow n}{f(a,k); f(c,k) \rightarrow n}\,Res}{f(c,k) \rightarrow n}\,\mathsf{Cut}_{f(x,k)}}{\rightarrow n}\,\mathsf{Cut}_{f(x,k)}$$

Our solution consists in designing another inference system, along the same ideas as before, for which Lemmas 5 and 6 still hold. To do so, we memorize more information in the mark (typically the constraints that need to be satisfied) so that the matching rule (removing the mark) can be applied only if the actual clauses would satisfy the constraints recorded in the mark.

Example 8. To explain the main idea, we give a simplified example of how the new proof system works. Coming back to Example 7, in our system we gety:

$$\dfrac{\rightarrow a \quad f(a,k) \rightarrow f(b,k)}{\rightarrow_{f(a,b), f(b,k), f(c,k) \notin X} f(b,k)}\,\mathsf{Cut}^1_{f(y,k)}$$

But we cannot apply Cut^2 since its application requires that the context satisfies the constraint in the mark, which is not the case. We could apply a Cut^1, without removing the mark but then the mark could not be removed any more since the marks can never be removed from the "pattern premisse" of a Cut^i_w rule.

If the clause is less constrained, for instance assume that we only impose $f(b,k) \notin X$, then we can prove $\to n$ as follows:

$$\dfrac{\to c \qquad \dfrac{\dfrac{\to a \quad f(a,k) \to f(b,k)}{\to_{f(b,k)\notin X} f(b,k)}\,\mathsf{Cut}^1_{f(y,k)} \qquad f(b,k); f(c,k) \to n}{f(c,k) \to n}\,\mathsf{Cut}^2}{\to n}\,\mathsf{Cut}^2_{f(y,k)}$$

This time, we may remove the mark, as the instance of X is the singleton $\{f(c,k)\}$, that does not contain $f(b,k)$.

We get an analog of Lemmas 5 and 6, which yields a PTIME decision procedure (because the number of possible marks is fixed).

Theorem 3. *If $\mathcal{S}$ is a set of ground clauses built on $\vdash$, we can decide in PTIME the satisfiability of $\mathcal{S}$ together with T, W and finitely many constrained clauses c_s, c_c built on the same pattern p, provided the constraints are monotone.*

Again, this can be extended, as in the theorem 2, guarding the clauses with predicates that are defined by a saturated set of Horn clauses $\mathcal{A}_0$ (w.r.t. a basic ordering). This can be extended also to the case where $\mathcal{S}$ contains equalities.

6 Conclusion

We designed a technique for proving tractability of a collection of proof systems (or Horn clauses): the idea is to extend the proof system with marked clauses such that the expressivity is unchanged while the unit strategy becomes complete. Our technique captures a class of clauses relevant to a computer security application.

PTIME membership is obtained by nesting PTIME oracles. We did not succeed however in showing a more abstract combination result allowing, say, to combine two tractable inference systems, one of which depends on the other. For instance, when we add guards to another system (resp. equalities in the input clauses) we would like to get automatically a tractability property from the tractability of the system without guards (resp. without equality) and the tractability of the guards entailment (resp. tractability of the word problem).

Another perspective is to provide a more abstract statement of the proof method, which does not rely on the specific deducibility predicate. Moreover, our work is not fully complete since we did not consider the function and reflexivity axioms in the two last sections. We could also investigate the case of several patterns and/or constraints that involve both a (non-ground) term and a set.

References

1. M Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

2. M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable Dolev-Yao style cryptographic library. In *17th IEEE Computer Science Foundations Workshop (CSFW'04)*, pages 204–218, 2004.
3. G. Bana, P. Adao, and H. Sakurada. Computationally complete symbolic attacker in action. In *32nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'12)*, pages 546–560, 2012.
4. G. Bana and H. Comon-Lundh. Towards unconditional soundness: Computationally complete symbolic attacker. In *1st International Conference on Principles of Security and Trust (POST'12)*, volume 7215 of *LNCS*, pages 189–208, 2012.
5. G. Barthe, B. Grégoire, S. Heraud, and S. Zanella Béguelin. Computer-aided security proofs for the working cryptographer. In *Advances in Cryptology (CRYPTO'11)*, volume 6841 of *LNCS*, pages 71–90. Springer, 2011.
6. D. Basin and H. Ganzinger. Automated complexity analysis based on ordered resolution. *J. of the Association of Computing Machinery*, 48(1):70–109, 2001.
7. B. Blanchet. A computationally sound mechanized prover for security protocols. In *IEEE Symposium on Security and Privacy (S&P'06)*, pages 140–154, 2006.
8. H. Comon and R. Treinen. The first-order theory of lexicographic path orderings is undecidable. *Theoretical Computer Science*, 176(1-2):67–87, April 1997.
9. A. Datta, A. Derek, J.C.Mitchell, and B.Warinschi. Computationally sound compositional logic for key exchange protocols. In *19th IEEE Computer Security Foundations Workshop (CSF'06)*, pages 321–334, 2006.
10. David McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2), 1993.
11. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Theory of Cryptography Conference (TCC 2004)*, volume 2951 of *LNCS*, pages 133–151, 2004.
12. R. Nieuwenhuis and A. Rubio. *Handbook of Automated Reasoning*, chapter Paramodulation-Based Theorem Proving. Elsevier Science and MIT Press, 2001.

A Proof of lemma 5

Lemma 5. *Let $\mathcal{S}$ be a set of ground clauses, and s be a ground term. In case $E = \emptyset$ and removing the function and reflexivity axioms from $\mathcal{I}$, $\mathcal{S} \Vdash_{\mathcal{I}}\rightarrow s$ if and only if $\mathcal{S} \Vdash_{\mathcal{J}}\rightarrow s$.*

Proof. We first prove that, if there is a proof Π of s in $\mathcal{I}$ from $\mathcal{S}$, then there is a proof Π' without W. Indeed, we may push W to the bottom of the proof as follows:

$$\dfrac{\dfrac{A_1, \ldots, A_n \rightarrow x}{A_1, \ldots, A_n, C \rightarrow x}\,W \qquad B_1, \ldots, B_m, w \rightarrow p}{A_1, \ldots, A_n, B_1, \ldots, B_m, C \rightarrow p}\,\mathsf{Cut}_w$$

can be rewritten to

$$\dfrac{\dfrac{A_1, \ldots, A_n, \rightarrow x \quad B_1, \ldots, B_m, w \rightarrow p}{A_1, \ldots, A_n, B_1, \ldots, B_m, \rightarrow p}\,\mathsf{Cut}_w}{A_1, \ldots, A_n, B_1, \ldots, B_m, C \rightarrow p}\,W$$

W also commutes with the rules Str_u. Since the proof of a unit clause cannot end with W, Π does not contain W.

Now let us show that if there is a proof of $\to s$ in $\mathcal{J}$ then there is a proof of $\to s$ in $\mathcal{I}$: Consider a minimal (in number of Cut^1, Cut^1_w, Str^1_u rules) proof Π of $S \to t$ in $\mathcal{J}$. Consider a subproof Π' of Π that uses once Cut^2_w, as a last inference rule. We show that Π' can be rewritten into a strictly smaller proof (w.r.t. the size). This contradicts the minimality of Π, hence this proves that the minimal size proof does not make use of any extra rule.

First note that, according to labels inheritance, once a clause is annotated, then the label cannot be removed completely, unless we apply Cut^2_w or Cut^2. Since the leaves of Π' are not annotated, we can write Π' as :

$$\dfrac{\dfrac{\vdots \qquad \dfrac{\vdots \quad \overset{\pi_1}{S^1 \to t}}{}\,R^1 \qquad \vdots}{S^n \to_p t}\,R^n \qquad \overset{\pi_2}{S, w\sigma \to p\sigma}}{S^n, S \to p\sigma}\,\mathsf{Cut}^2_w$$

where $R^1, \ldots, R^n$ are Cut^1_w, Cut^1 or Str^1_u.

We argue that Π' can be rewritten into

$$\dfrac{\vdots \qquad \dfrac{\vdots \qquad \dfrac{\overset{\pi_1}{S^1 \to t} \qquad \overset{\pi_2}{S, w\sigma \to p\sigma}}{S^1, S \to p\sigma}\,\mathsf{Cut}^2_w}{}\,\widetilde{R^1} \qquad \vdots}{S^n, S \to p\sigma}\,\widetilde{R^n}$$

This is a strictly smaller proof. It only remains to define the rules $\widetilde{R^i}$ and check that the above proof is a valid proof in the new inference system indeed.

$$\text{If} \qquad R^k = \frac{V_2^k \ \to_p t^k \quad V_1^k, w'\sigma \ \to_p t}{S^k \ \to_p t}$$

$$\text{We let} \qquad \widetilde{R_k} = \frac{V_2^k \ \to_p t^k \quad S, V_1^k, w'\sigma \to p\sigma}{S, S^k \to p\sigma}$$

The rule $\mathsf{Cut}^1_{w'}$ is therefore replaced with a rule $\mathsf{Cut}^2_{w'}$.

$$\text{If} \qquad R^k = \frac{V_1^k, v\sigma \to_p t}{V_1^k \to_p t} \qquad \text{we let} \qquad \widetilde{R^k} = \frac{S, V_1^k, v\sigma \to p\sigma}{S, V_1^k \to p\sigma}$$

The rule Str^1_v is replaced with a rule Str^2_v.

It is now enough to note that the choice of $\widetilde{R^k}$ ensures that Π' is a valid proof in the $\mathcal{I}$ inference system.

B Proof of tractability of deducibility axioms

B.1 Adding equalities

Let us recall the transitivity rule:

$$\frac{X \vdash x \quad X; z \vdash y \quad x =_E z}{X \vdash y} \, T(E)$$

We define the unit resolution rule with equalities $Res_u(E)$ as follows, where the predicate p ranges over $=$ and $\vdash$

$$\frac{p(t_1, t_2) \quad p(u_1, u_2), A_1, \cdots, A_n \to B}{A_1, \cdots, A_n \to B} \, t_1 =_E u_1, t_2 =_E u_2$$

We also define an equality elimination rule $\textsc{Elim}(E)$ as follows :

$$\frac{t_1 = t_2, A_1, \cdots, A_n \to B}{A_1, \cdots, A_n \to B} \, t_1 =_E t_2$$

Lemma 7. *Given a list of terms $S_1 \vdash u_1, \cdots, S_n \vdash u_n$, a finite set of ground equations E and a goal term $S \vdash u$, the problem $E, S_1 \vdash u_1, \cdots, S_n \vdash u_n, (x \vdash x)_{x \in \mathcal{T}} \Vdash^?_{W,T(E)} S \vdash u$ is decidable in polynomial time with access to an oracle deciding E.*

Proof. Let T be the set of terms occurring in $S_1, \ldots, S_n, S, u_1, \ldots, u_n, u$. Split T into disjoint equivalence classes modulo E, calling (at most) a quadratic number of times the oracle deciding E. For each equivalence class, choose a representative and replace the terms in $S_1 \vdash u_1, \cdots, S_n \vdash u_n, S \vdash u$ with their representatives. Then, the resulting entailment problem can be turned into a HornSat problem (as before), replacing every representative of an equivalence class with a proposition variable.

We then get the main lemma :

Lemma 3. *Given a set of ground Horn clauses (built on $\vdash$ and $=$) $\mathcal{C}$, the satisfiability of $\mathcal{C} \cup \{r, w, t\}$ is decidable in polynomial time.*

Proof. Let $\mathcal{B}_1$ be the set of occurring in $\mathcal{C}$. Let $\mathcal{B}_2$ be the set of $S \vdash t$ occurring in $\mathcal{C}$, as well as $\perp$.

We want to compute the least fixed point of the following function F which takes as input a set of equations $E \subseteq \mathcal{B}_1$ and a set of $\vdash$ literals $\mathcal{B}$ and returns E' and $\mathcal{B}'$ built as follows :

$$\mathcal{B}' = \{S \vdash t \in \mathcal{B}_2 | \mathcal{C} \cup \mathcal{B} \Vdash_{Res_u(E), \textsc{Elim}(E)} S \vdash t\} \cup \{S \vdash t \in \mathcal{B}_2 | \mathcal{B} \Vdash_{R,W,T(E)} S \vdash t\}$$

$$E' = \{u = v \in \mathcal{B}_1 | \mathcal{C} \cup \mathcal{B} \Vdash_{Res_u(E), \textsc{Elim}(E)} u = v\}$$

Our algorithm answers $\textsc{Unsatisfiable}$ iff $\perp$ is derivable using from the least fixed point of F using unit resolution.

E is always a finite (polynomially bounded) set of ground equations. Hence there is an polynomial time oracle that decides the equality modulo E, for instance using a congruence closure algorithm. Then, thanks to lemma 7, F can be computed in polynomial time. Furthermore, the number of iterations of F is linear. Hence the fixed point cand be computed in polyniomial time.

Let $\mathcal{B}_f, E_f$ be the least fixed point of F. Let us prove now that $\mathcal{C} \cup \{r, w, t\}$ is satisfiable iff $\perp \notin \mathcal{B}_f$.

If $\perp \in \mathcal{B}_f$, then $\mathcal{C} \cup \{r, w, t\}$ is unsatisfiable since every deduction step used in the computation of F is a consequence of $\mathcal{C} \cup \{r, w, t\}$ (and the equality axioms).

Conversely, if $\perp \notin \mathcal{B}_f$, we consider the first-order structure $\mathcal{M}$, in which the interpretation domain is the quotient $\mathcal{T}/ =_{E_f}$ of the set of ground terms by the congruence generated by E_f and the interpretation of $\vdash$ is the set $\{S \vdash t \mid \mathcal{B}_2 \Vdash_{R,W,T(E_f)} S \vdash t\}$.

$\mathcal{M}$ is, by construction, a model of r, w, t. If $S_1 \vdash t_1, \ldots, S_n \vdash t_n, u_1 = v_1, \ldots, u_m = v_m \rightarrow H$ is a clause of $\mathcal{C}$, and, for every i, $\mathcal{M} \models S_i \vdash t_i$ and, for every j, $u_j =_{E_f} v_j$, then, for every i, $\mathcal{B}_f \Vdash_{R,W,T(E_f)} S \vdash t_i$. Hence $S_i \vdash t_i$ is in the first component of $F(\mathcal{B}_f, E_f)$, hence in $\mathcal{B}_f$.

It follows that, $\mathcal{B}_f, \mathcal{C} \Vdash_{Res_u(E),\textsc{Elim}(E)} H$. Hence $H \neq \perp$ and H is in either components of $F(\mathcal{B}_f, E_f) = (\mathcal{B}_f, E_f)$. Therefore $\mathcal{M} \models H$.

We have proved that $\mathcal{M}$ is a model of each clause of $\mathcal{C}$. Since it is a model of t, w, r, this concludes the proof.

B.2 Adding a function axiom

First, note that the axiom $F_f : S \vdash x_1, \cdots, S \vdash x_n \rightarrow S \vdash f(x_1, \cdots, x_n)$ is equivalent (modulo weakening and transitivity) to $t_1; \ldots; t_{n_f} \vdash f(u_1, \cdots, u_{n_f})$

Now we know by lemma **??** that $\mathcal{C} \cup \{r, w, t\} \cup f(\mathcal{F})$ is satisfiable if and only if the empty clause is not derivable from $\mathcal{C} \cup \{t_1; \ldots; t_{n_f} \vdash f(u_1, \cdots, u_{n_f}) | f \in \mathcal{F}, u_1, \ldots, u_{n_f} \in \mathcal{T}\}$ with the rules $T(E)$, $Res_u(E)$, $\textsc{Elim}(E)$ and W.

Lemma 8. *Given a set of ground equations E, a set of ground terms $u, t_1, \ldots, t_n$, and $\mathcal{F}$ a set of function symbols. The problem $\exists C.C[t_1, \ldots, t_n] = u$ with C (multi)context built on $\mathcal{F}$ is decidable in polynomial time.*

Proof. We proceed as follows (the steps will be precised later) :

1. Build a tree automaton $\mathcal{A}$ (of polynomial size) that recognizes the set of all t such that $t =_E u$.
   (a) Compute (in polynomial time in $|E|$) a flat convergent rewriting system R for E (of polynomial size in the size of E).
   (b) Build a tree automaton (of polynomial size in $|R| + |t|$), which accepts the terms that rewrite to $t\downarrow_R$.
2. Build a tree automaton $\mathcal{B}$ (of polynomial size in $\Sigma_i |t_i|$) that recognizes the language $\{C[t_1, \ldots, t_n]\}$.
3. Check (in polynomial time in $|\mathcal{A}| + |\mathcal{B}|$) whether $L(\mathcal{A}) \cap L(\mathcal{B}) = \emptyset$.

1a - We add a constant c_u for every subterm u of E, t and add an equation $c_u = u$ to E. In this way, we may now assume w.l.o.g that every equation in E has the form $f(a_1, \ldots, a_n) = a$ or $a_1 = a$. We choose an arbitrary linear order on symbols in which the non-constant function symbols are greater than the constants and run a Knuth-Bendix completion on E using a lexicographic path ordering that extends this precedence. This yields a flat convergent rewrite system R whose size is polynomial in E. This requires only a polynomial time.

1b - We want to recognize the set of terms u such that $u{\downarrow_R} = t{\downarrow_R}$. Note that $t{\downarrow_R}$ is a constant c_t. Now build a tree automaton $\mathcal{A}$ as follows :

- the set of states of $\mathcal{A}$ is the set S of constants appearing in R,
- for each constant c add a transition $c() \to c$
- for each rule $f(a_1, \ldots, a_n) \to a$ in R add a transition $f(a_1, \ldots, a_n) \to a$ in $\mathcal{A}$
- for each rule $a_1 \to a$ in R and every transition $f(a_1, \ldots, a_n) \to a_1$ replace a_1 by a in the transition (applying this point starting from the highest a_1 in the order chosen to complete E).
- the accepting state of $\mathcal{A}$ is c_t

Note that this procedure yields a polynomial size $\mathcal{A}$ in polynomial time. If $\mathcal{A}$ recognizes u, it is clear that the accepting run of $\mathcal{A}$ can be seen as a rewrite sequence from u to c_t. Conversely, each accepting run on a term u yields a rewrite sequence from u to c_t.

2 - Build the tree automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ recognizing the terms $t_1, \ldots, t_n$ with accepting state q_0. Now let $\mathcal{A}'$ be the automaton recognizing the language $t_1, \ldots, t_n$ with accepting state q_0 (it is the sum of the n previous automata). Let $\mathcal{B}$ be the automaton obtained extending $\mathcal{A}'$ with the transitions $f(q_0, \ldots, q_0) \to q_0$. Note that $\mathcal{B}$ is built in polynomial time and is of polynomial size. It is clear that $\mathcal{B}$ recognizes the language $\{C[t_1, \ldots, t_n] |\ C$ context built on $\mathcal{F}\}$.

3 - Build the product automaton that recognizes $L(\mathcal{A}) \cap L(\mathcal{B})$ (of polynomial size) and test for emptiness in polynomial time.

Lemma 9. *Given a list of terms $S_1 \vdash u_1, \cdots, S_n \vdash u_n$, a finite set of ground equations E and a goal term $S \vdash u$, the problem $S_1 \vdash u_1, \cdots, S_n \vdash u_n, (t_1; \ldots; t_{n_f} \vdash f(t_1, \cdots, t_{n_f}))_{f \in \mathcal{F}, t_1, \ldots, t_{n_f} \in \mathcal{T}} \Vdash^?_{W,T(E)} S \vdash u$ is decidable in polynomial time.*

Proof. Note that in the proof of lemma 7 we saturate $S_1 \to u_1 \ldots S_n \to t_n, S$, modulo the unit verion of $T(E)$ and check if we obtain u. Now, we need to saturate $S_1 \to u_1 \ldots S_n \to t_n, S, \neg u, (f(t_1, \cdots, t_{n_f}))_{f \in \mathcal{F}, t_1, \ldots, t_{n_f} \in \mathcal{T}}$ modulo the unit version of $T(E)$. Observe the following : if a function clause is used to derive u then it is used in a proof that has the following structure (we omit here that everything is done modulo E)

$$\dfrac{\dfrac{t_i \qquad \dfrac{\dfrac{\dfrac{t_l \qquad t_1,\ldots,t_n \to f(t_1,\ldots,t_n)}{t_1,\ldots,t_{l-1},t_{l+1},\ldots,t_n \to f(t_1,\ldots,t_n)}}{\cdots}}{t_i \to f(t_1,\ldots,t_n)}}{f(t_1,\ldots,t_n)} \qquad f(t_1,\ldots,t_n),A_1,\ldots,A_k \to B}{A_1,\ldots,A_k \to B}$$

In its turn either t_i is a term in $U = \bigcup_i S_i \cup S \cup \{t_1,\ldots,t_n\}$ or its proof has the structure shown above. Therefore, there exists $v_1,\ldots,v_l,w \in U$ (and the units $v_1,\ldots,v_l$ are derivable) such that $w = f(t_1,\ldots,t_n)$ and $C[v_1,\ldots,v_l] =_E w$. Note that this observation gives us the following :

$$E, \quad S_1 \vdash u_1, \cdots, S_n \vdash u_n, \qquad (t_1;\ldots;t_{n_f} \vdash f(t_1,\cdots,t_{n_f}))_{f\in\mathcal{F},t_1,\ldots,t_{n_f}\in\mathcal{T}} \Vdash^{?}_{W,T(E)} S \vdash u$$

is decidable in PTIME by saturating $E, S_1 \to u_1, \cdots, S_n \to u_n, S$ by

$$\frac{x \qquad X, z \to y}{X \to y}\ \exists C.C[t_1,\ldots,t_n] =_E z$$

(where $t_1,\ldots,t_n$ are units provable with the previous rule) and checking whether $\exists C.C[v_1,\ldots,v_k] =_E u$ where $v_1,\ldots,v_k$ are the units derived by the saturation. As checking the condition $\exists C.C[t_1,\ldots,t_n] =_E u$ is decidable in PTIME, the saturation in in PTIME.

Lemma 4. *Given a set of ground Horn clauses (built on $\vdash$ and $=$) $\mathcal{C}$, the satisfiability of $\mathcal{C} \cup \{r,w,t\} \cup f(\mathcal{F})\}$ is decidable in polynomial time.*

Proof. The proof goes exactly as the proof of lemma 3 except that we use the oracle of lemma 9 instead of the oracle of lemma 7.

C Proof for section 4

Recall the inference rules:

$$\frac{A_1,\ldots,A_n \to_? x \qquad B_1,\ldots,B_m, w \to_? v}{A_1,\ldots,A_n,B_1,\ldots,B_m \to_p w}\ \mathsf{Cut}^1_w$$

$$\frac{A_1,\ldots,A_n \to_? y \qquad w, B_1,\ldots,B_m \to p}{A_1,\ldots,A_n,B_1,\ldots,B_m \to p}\ \mathsf{Cut}^2_w$$

$$\frac{A_1,\ldots,A_n \to_? x \qquad B_1,\ldots,B_m, x \to_? v}{A_1,\ldots,A_n,B_1,\ldots,B_m \to_? v}\ \mathsf{Cut}^1$$

in which the conclusion is marked iff one of the premises is marked.

$$\frac{A_1,\dots,A_n \to_? x \qquad x, B_1,\dots,B_m \to p}{A_1,\dots,A_n,B_1,\dots,B_m \to p}\,\mathsf{Cut}^2$$

$$\frac{A_1,\dots,A_n,u \to_? x}{A_1,\dots,A_n \to_p x}\,\mathsf{Str}^1_u \qquad \frac{A_1,\dots,A_n,u \to_? p}{A_1,\dots,A_n \to p}\,\mathsf{Str}^2_u$$

Recall lemma 6

Lemma 6. *If $\mathcal{S} \Vdash_{\mathcal{J}} \to s$ then $\to s$ is derivable from $\mathcal{S}$ in $\mathcal{J}$ using the unit strategy.*

Proof. Let Π be a proof of $\to s$ in $\mathcal{J}$ minimal in the number of non unit cuts. Assume, by contradiction that Π uses at least one non-unit rule, for example the following instance of Cut^2_w,

$$R^0\frac{S \to_p u \quad S', w\sigma \to p\sigma}{S, S' \to p\sigma}$$

then as the conclusion of Π is a unit clause, Π has a subproof of the following form :

$$\cfrac{\begin{array}{c}\cfrac{\cfrac{}{S^0 \to p\sigma}\,R^0}{S^1 \to p\sigma}\,R^1 \\ \hline \\ \vdots \end{array}}{\to p\sigma}\,R^n$$

Let $I = \{i_1,\dots,i_l\}$ be the set of indices such that $S^i \backslash S^{i-1} \subseteq S$. If $i \in I$ and

$$R^i\frac{\to_? t^i \quad S^{i-1} \to p\sigma}{S^i \to p\sigma}$$

we let

$$\widetilde{R^i}\frac{\to_? t^i \quad S \cap S^{i-1} \to_p u}{S \cap S^i \to_p u}$$

and if $i \in I$ and

$$R^i\frac{S^{i-1} \to p\sigma}{S^i \to p\sigma}$$

we let

$$\tilde{R}^i\frac{S^{i-1} \to_p u}{S^i \to_p u}$$

Then replacing the original subproof by the following one in Π yields a proof with one less non-unit cut.

$$\dfrac{\dfrac{\dfrac{S \to_p u}{\cdots}\,\widetilde{R^{i_1}}}{\to_p u}\,\widetilde{R^{i_l}} \qquad S', w\sigma \to p\sigma}{S' \to p\sigma}$$

Theorem 1. *If $\mathcal{S}$ is a set of ground clauses built on $\vdash$, we can decide in PTIME the satisfiability of $\mathcal{S}$, together with T, W and finitely many clauses c_n, c_s, c_c, that are built on the same pattern p.*

Proof. First observe that the unit resolution strategy in 6 yields a PTIME decision procedure for the problem : $\mathcal{S} \Vdash_{\mathcal{J}} \to s$. Now to solve , $\mathcal{S} \Vdash_{\mathcal{J}} S \to s$ observe that it is enough to erase the elements of S in all premises of clauses in $\mathcal{S}$ (yielding $\mathcal{S}'$) and check if $\mathcal{S}' \Vdash_{\mathcal{J}} \to s$ which is decidable in PTIME.

Now we only have to use the previous oracle instead of the one of lemma 7 in the proof of lemma 3 yielding our theorem.

D Proof of the general case

Consider rules :

$$\textsc{Cut}_i \dfrac{X \to y \quad X', u_i(y) \to p(x)}{X; X' \to p(x)}\ \Gamma_i(X \cup X')$$

and

$$\textsc{Str}_i \dfrac{X; v_i(y) \to p(x)}{X \to p(x)}\ \Delta_i(X)$$

We consider now labeled clauses $S \ {}_{\Delta}{\to}_{\Gamma}\ t$ where Δ, Γ are finite sets of constraints.

We add the following rules :

$$\textsc{Cut}_i^1 \dfrac{X \ {}_{\Delta_1}{\to}_{\Gamma_1}\ y \quad X', u_i(y) \ {}_{\Delta_2}{\to}_{\Gamma_2}\ x}{X, X' \ {}_{\Delta_2}{\to}_{\Gamma_1, \Gamma_2, \Gamma_i}\ x}\ \Gamma_i(X \cup X'),\ \Gamma_i \in \Delta_1,\ \Gamma_1(X')$$

$$\textsc{Cut}_i^2 \dfrac{X \ {}_{\Delta_1}{\to}_{\Gamma_1}\ y \quad X', u_i(y) \to p(x)}{X, X' \to p(x)}\ \Gamma_i(X \cup X'),\ \Gamma_i \in \Delta_1,\ \Gamma_1(X')$$

$$\textsc{Str}_i^1 \dfrac{X, v_i(y) \ {}_{\Delta}{\to}_{\Gamma}\ x}{X \ {}_{\Delta}{\to}_{\Gamma \cup \Delta_i}\ x}\ \Delta_i(X)$$

$$(\textsc{Context}) \dfrac{X \to x}{X \ {}_{\Delta} \to x} \Delta \subseteq \{\Gamma | \Gamma(X)\}$$

Lemma 10. *The previous* $\textsc{Cut}_i^1, \textsc{Cut}_i^2, \textsc{Str}_i^1, (\textsc{Context})$ *are sound and complete with respect to* $\textsc{Cut}_i, \textsc{Str}_i$ *cut and weakening.*

Proof. Assume that $\mathcal{S}$ is a set of Horn clauses (without annotations with constraint sets) and that $S \to t$ is a clause, that is derivable in the inference system that includes the new extra rules. We show below that $S \to t$ is also provable without the extra rules.

We first note that, if one of the premisses of a rule has a non-empty left or right constraint, it is also the case of the conclusion, except for the rule $\textsc{Cut}_i^2$. Therefore, any proof of a clause $S \to t$ that uses one of the additional rules, must also use at least once $\textsc{Cut}_i^2$. Consider a minimal (in size) proof Π of $S \to t$, that might use the extra rules. Consider a subproof Π' of Π that uses once $\textsc{Cut}_i^2$, as a last inference rule. We show that Π' can be rewritten into a strictly smaller proof (w.r.t. the size). This contradicts the minimality of Π, hence this proves that the minimal size proof does not make use of any extra rule.

First note that, according to labels inheritance, once a clause is annotated with sets of constraints, then the labels cannot be removed completely, unless we apply $\textsc{Cut}_i^2$. Since the leaves of Π' are not annotated with sets of constraints, we can write Π' as :

$$\dfrac{\dfrac{\begin{array}{cc} \vdots & \begin{array}{c} \dfrac{\vdots \qquad \dfrac{\overset{\pi_1}{S^1 \to t}}{S^1 \;{}_{\Delta}{\to}_{\emptyset}\; t}\;\Delta_1^1(S_1)}{}\;R^1 \\ \vdots \end{array} \end{array}}{S^n \;{}_{\Delta_1^n}{\to}_{\Gamma_1^n}\; t}\;R^n \qquad \overset{\pi_2}{S, u_i(t) \to p(u)}}{S^n, S \to p(u)}\;\Gamma_i(S^n, S), \Gamma_i \in \Delta_1^n, \Gamma_1^n(S)$$

where π_1, π_2 are proofs that do not use the extra rules and $R^1, \ldots, R^n$ are in $\textsc{Cut}_i^1, \textsc{Str}_i^1$. In particular, $\Gamma_1^1 \subseteq \ldots \subseteq \Gamma_1^n$ since these two rules only increase the right set of constraints.

We argue that Π' can be rewritten into

$$\dfrac{\begin{array}{cc} \vdots & \begin{array}{c} \dfrac{\vdots \qquad \dfrac{\overset{\pi_1}{S^1 \to t} \qquad \overset{\pi_2}{S, u_i(t) \to p(u)}}{S^1, S \to p(u)}\;\Gamma_i(S^1, S)}{}\;\widetilde{R^1} \\ \vdots \end{array} \end{array}}{S^n, S \to p(u)}\;\widetilde{R^n}$$

This is a strictly smaller proof, which is what we want. It only remains to define the rules $\widetilde{R^i}$ and check that the above proof is a valid proof in the new inference system indeed.

If

$$R^k = \frac{V_2^k \ {}_{\Delta_2^k} \rightarrow_{\Gamma_2^k} t^k V_1^k, u_i(t^k) \ {}_{\Delta_1^k} \rightarrow_{\Gamma_1^k} t}{S^k \ {}_{\Delta_1^k \cap \Delta_2^k} \rightarrow_{\Gamma_1^k, \Gamma_2^k, \Gamma_k} t} \Gamma_k(V_1^k, V_2^k),\ \Gamma_k \in \Delta_2^k,\ \Gamma_2^k(V_1^k)$$

we let

$$\widetilde{R_k} = \frac{V_2^k \ {}_{\Delta_2^k} \rightarrow_{\Gamma_2^k} t^k S, V_1^k, u_i(t^k) \rightarrow p(u)}{S, S^k \rightarrow p(u)} \Gamma_k(S, V_1^k, V_2^k),\ \Gamma_k \in \Delta_2^k,\ \Gamma_2^k(S, V_1^k)$$

The rule $\textsc{Cut}_i^1$ is therefore replaced with a rule $\textsc{Cut}_i^2$. The conditions are satisfied indeed (we get a valid proof):

- $\Gamma_k \in \Gamma_1^k \cup \Gamma_2^k \cup \{\Gamma_k\} = \Gamma_1^{k+1} \subseteq \Gamma_1^n$ and S satisfies Γ_1^n, hence $\Gamma_k(V_1^k, V_2^k) \implies \Gamma_k(S, V_1^k, V_2^k)$
- $\Gamma_2^k \subseteq \Gamma_1^{k+1} \subseteq \Gamma_1^n$, and S satisfies Γ_1^n, hence $\Gamma_2^k(V_1^k) \implies \Gamma_2^k(S, V_1^k)$

If

$$R^k = \frac{V_1^k, v_i(u^k) \ {}_{\Delta_1^k}\!\rightarrow_{\Gamma_1^k} t}{V_1^k \ {}_{\Delta_1^k}\!\rightarrow_{\Gamma_1^k \cup \{\Delta_i\}} t} \Delta_i(V_1^k)$$

we let

$$\widetilde{R^k} = \frac{S, V_1^k, v_i(u^k) \rightarrow t}{S, V_1^k \rightarrow t} \Delta_i(S, V_1^k)$$

The rule $\textsc{Str}_i^1$ is replaced with a rule $\textsc{Str}_i$. The condition is satisfied since , as before, $\Delta_i \in \Gamma_1^n$ and S satisfies Γ_1^n.

In order to have a unit strategy, we need to modify the rules a little. Intuitively, the multiset L stands for a set of cuts that are done in advance. Note that remembering the precise cut is not usefull, it is enough to remember the constraints involved in the cut.

$$\textsc{Cut}_i^1 \frac{X \ {}_{\Delta_1} \rightarrow_{\Gamma_1}^{L_1} y X', u_i(y) \ {}_{\Delta_2} \rightarrow_{\Gamma_2}^{L_2} x}{X, X' \ {}_{\Delta_1 \cap \Delta_2} \rightarrow_{\Gamma_1, \Gamma_2, \Gamma_i}^{L_1, L_2} x} \begin{array}{l} \Gamma_i(X \cup X'),\ \Gamma_i \in \Delta_1,\ \Gamma_1(X'), \\ \forall(\neg\Gamma \rightarrow \neg\Delta) \in L_1, L_2 . \Gamma_i, \Gamma_1 \notin \Delta \end{array}$$

$$\textsc{Cut}_i^2 \frac{X \ {}_{\Delta_1}\!\rightarrow_{\Gamma_1}^{L_1} y \quad X', u_i(y) \rightarrow^{L_2} p(x)}{X, X' \rightarrow^{L_1, L_2} p(x)} \begin{array}{l} \Gamma_i(X \cup X'),\ \Gamma_i \in \Delta_1, \Gamma_1(X'), \\ \forall(\neg\Gamma \rightarrow \neg\Delta) \in L_1, L_2 . \Gamma_i, \Gamma_1 \notin \Delta \end{array}$$

$$\textsc{Str}_i^1 \frac{X, v_i(y) \ {}_{\Delta}\!\rightarrow_{\Gamma}^{L_1} x}{X \ {}_{\Delta}\!\rightarrow_{\Gamma \cup \Delta_i}^{L_1} x} \Delta_i(X),\ \forall(\neg\Gamma \rightarrow \neg\Delta') \in L_1 . \Delta_i \notin \Delta'$$

$$(\textsc{Context}) \frac{X \rightarrow^L x}{X \ {}_{\Delta} \rightarrow^L x} \Delta \subseteq \{\Gamma | \Gamma(X),\ \forall(\neg\Gamma \rightarrow \neg\Delta) \in L . \Gamma \notin \Delta\}$$

we also add :

$$(1'_p)_i \frac{{}_{\Delta_1} \rightarrow^{L_1}_{\Gamma_1} yX', u_i(y) \;\; {}_{\Delta_2} \rightarrow^{L_2}_{\Gamma_2} x}{X' \;\; {}_{\Delta_1 \cap \Delta_2} \rightarrow^{L_1, L_2, (\neg \Gamma_1, \Gamma_i \rightarrow \neg \Delta(u_i(y))^c)}_{\Gamma_2} x} \Gamma_i \in \Delta_1$$

$$(1''_p)_i \frac{{}_{\Delta_1}\rightarrow^{L_1}_{\Gamma_1} y \quad X', u_i(y) \rightarrow^{L_2} p(x)}{X' \rightarrow^{L_1, L_2, (\neg \Gamma_1, \Gamma_i \rightarrow \neg \Delta(u_i(y))^c)} p(x)} \Gamma_i \in \Delta_1$$

$$(2'_p)_i \frac{X, v_i(y) \;\; {}_{\Delta}\rightarrow^{L_1}_{\Gamma} x}{X \;\; {}_{\Delta}\rightarrow^{L_1, (\neg \Delta_i \rightarrow \neg \Delta(u_i(y))^c)}_{\Gamma} x}$$

$$(\textsc{Remove}_1) \frac{X \;\; {}_{\Delta_1}\rightarrow^{L, (\neg \Gamma \rightarrow \neg \Delta)}_{\Gamma_1} x}{X \;\; {}_{\Delta_1}\rightarrow^{L}_{\Gamma_1, \Gamma} x} \Gamma(X), \quad \forall (\neg \Gamma' \rightarrow \neg \Delta') \in L. \Gamma \notin \Delta$$

$$(\textsc{Remove}_2) \frac{X \rightarrow^{L, (\neg \Gamma \rightarrow \neg \Delta)} p(x)}{X \;\; \rightarrow^{L} p(x)} \Gamma(X), \quad \forall (\neg \Gamma' \rightarrow \neg \Delta') \in L. \Gamma \notin \Delta$$

We now have to show that adding these new rules is sound.

Lemma 11. *If $\mathcal{S}$ entails $S \;{}_{(\Delta)}\!\rightarrow_{(\Gamma)} t$ with the modified rules then $\mathcal{S}$ entails $S \;{}_{(\Delta)}\!\rightarrow_{(\Gamma)} t$ with rules* $\textsc{Cut}_i, \textsc{Str}_i, \textsc{Cut}^1_i, \textsc{Cut}^2_i, \textsc{Str}^1_i$ *and* $\textsc{Context}$*, cut and weakening.*

Proof. First of all, note that if a proof Π with the new rules does not use $(1'_p)_i, (2'_p)_i, (1''_p)_i$ then for all clause $S \;{}_{(\Delta)}\!\rightarrow^{L}_{\Gamma)} t$ in Π, L is empty. Note that with L empty, the old and the new version of $\textsc{Cut}^1_i, \textsc{Cut}^2_i, \textsc{Str}^1_i$ are the same as the old ones, therefore, Π is a valid proof in the old proof system.

Let Π be a proof of $S \;{}_{(\Delta)}\!\rightarrow_{(\Gamma)} t$. Assume that the number of rules $(1'_p)_i, (2'_p)_i, (1''_p)_i$ is minimal in Π. By contradiction assume that there is a rule $(1'_p)_i, (2'_p)_i, (1''_p)_i$ in Π. Assume that it is the following $(1'_p)_i$ rule :

$$R^0 \frac{{}_{\Delta_1} \rightarrow^{L_1}_{\Gamma_1} uS, u_i(u) \;\; {}_{\Delta_2} \rightarrow^{L_2}_{\Gamma_2} v}{S \;\; {}_{\Delta_1 \cap \Delta_2} \rightarrow^{L_1, L_2, (\neg \Gamma_1, \Gamma_i \rightarrow \neg \Delta(u_i(u))^c)}_{\Gamma_2} v} \Gamma_i \in \Delta_1$$

As the conclusion of Π is not annotated by $(\neg \Gamma_1, \Gamma_i \rightarrow \neg \Delta(u_i(u))^c)$ there is in Π after the previous cut a $\textsc{Remove}$ rule of the following form – assume that it is a $\textsc{Remove}_1$ rule (the $\textsc{Remove}_2$ case is similar) – we take R^n as the first occurence of such a rule after R^0

$$R^n \frac{S' \;\; {}_{\Delta}\rightarrow^{L, (\neg \Gamma_1, \Gamma_i \rightarrow \neg \Delta(u_i(u))^c)}_{\Gamma} v'}{S' \;\; {}_{\Delta} \rightarrow^{L}_{\Gamma, \Gamma_i, \Gamma_1} v'} \Gamma_i, \Gamma_1(S'), \quad \forall (\neg \Gamma' \rightarrow \neg \Delta') \in L. \Gamma_i, \Gamma_1 \notin \Delta'$$

Let $R^1, \ldots, R^{n-1}$ be the path in Π from the R^0 to R^n. If R^i is

$$R^k \frac{S^k_1 \;\; {}_{\Delta_1}\rightarrow^{L^k_1}_{\Gamma_1} u^k S^k_2, u_i(u^k) \;\; {}_{\Delta_2}\rightarrow^{L^k_2}_{\Gamma_2} v^k}{S^k_1, S^k_2 \;\; {}_{\Delta_1 \cap \Delta_2}\rightarrow^{L^k}_{\Gamma_1, \Gamma_2, \Gamma_i} v^k} \begin{array}{l} \Gamma_i(S^k_1 \cup S^k_2),\ \Gamma_i \in \Delta_1,\ \Gamma_1(S^k_2), \\ \forall (\neg \Gamma \rightarrow \neg \Delta) \in L_1, L_2. \Gamma_i, \Gamma_1 \notin \Delta \end{array}$$

if R^{k-1} is the left premise of R^k, we take $\widetilde{R^k}$ as

$$\widetilde{R^k}\,\frac{S_1^k, u_i(u)\ {}_{\Delta_1}{\to}^{\widetilde{L^{k-1}}}_{\Gamma_1}\ u^k \quad S_2^k, u_i(u^k)\ {}_{\Delta_2}{\to}^{L_2^k}_{\Gamma_2}\ v^k}{S_1^k, S_2^k\ {}_{\Delta_1\cap\Delta_2}{\to}^{\widetilde{L^k}}_{\Gamma_1,\Gamma_2,\Gamma_i}\ v^k}\ \begin{array}{l}\Gamma_i(S_1^k,S_2^k,u),\ \Gamma_i\in\Delta_1,\ \Gamma_1(S_2^k),\\ \forall(\neg\Gamma\to\neg\Delta)\in L_1,L_2.\Gamma_i,\Gamma_1\notin\Delta\end{array}$$

With $\widetilde{L^k} = \widetilde{L^{k-1}}, L_2^k$. As $(\neg\Gamma_1, \Gamma_i \to \neg\Delta(u_i(u))^c)$ is in L_2, we know that $\Gamma_i, \Gamma_1 \notin \Delta(u_i(u))^c$, therefore, $\Gamma_i(u_i(u))$ holds and $\Gamma_1(u_i(u))$ holds, the constraints of $\widetilde{R^k}$ are satisfied.

We make the same transformation if R^k is any other rule, and the same argument gives the fact that these transformations are correct. Note that $\widetilde{L^k} = L^k \backslash (\{(\neg\Gamma_1, \Gamma_i \to \neg\Delta(u_i(u))^c)\} \cup L_1)$.

Now write :

$$\widetilde{R^n}\,\frac{{}_{\Delta_1}{\to}^{L_1}_{\Gamma_1}\ u \quad S', u_i(u)\ {}_{\Delta_2}{\to}^{\widetilde{L^{n-1}}}_{\Gamma_2}\ v'}{S'\ {}_{\Delta_1\cap\Delta_2}{\to}^{L_1,L_2}_{\Gamma_1,\Gamma_2,\Gamma_i}\ x}\ \begin{array}{l}\Gamma_i(S'),\ \Gamma_i\in\Delta_1,\ \Gamma_1(S'),\\ \forall(\neg\Gamma\to\neg\Delta)\in L.\Gamma_i,\Gamma_1\notin\Delta\end{array}$$

Let Π' be Π in which we remove R^0 and for $k = 1..n$ we substitute $\widetilde{R^k}$ for R^k. The inference Π' is a valid inference of $S\ {}_{(\Delta)}{\to}_{(\Gamma)}\ t$, with one less $(1'_p)_i, (2'_p)_i, (1''_p)_i$ rule than Π wich contradicts our hypothesis. We conclude, that there is no $(1'_p)_i, (2'_p)_i, (1''_p)_i$ in Π, therefore Π is an inference in the old inference system.

Lemma 12. *With the previous rules, a unit saturation strategy is complete.*

Proof. Let Π be a proof of $\to t$ with a minimal number of non unit rules. Let us assume, by contradiction that Π contains at least one non unit rule, let R^0 be a bottommost such rule, assume that R^0 is an instance of $\textsc{Cut}_i^2$

$$R^0\,\frac{S\ {}_{\Delta_1}{\to}^{L_1}_{\Gamma_1}\ u \quad S', u_i(u) \to^{L_2} p(v)}{S, S' \to^{L_1,L_2} p(v)}\ \begin{array}{l}\Gamma_i(S\cup S'),\ \Gamma_i\in\Delta_1,\ \Gamma_1(S'),\\ \forall(\neg\Gamma\to\neg\Delta)\in L_1,L_2.\Gamma_i,\Gamma_1\notin\Delta\end{array}$$

We know that all rules following R^0 are unit rules, so there is a path in Π such that :

$$\cfrac{\cfrac{\cfrac{}{S^0 \to^{L_1,L_2} p(v)}\,R^0}{S^1 \to^{L^1} p(v)}\,R^1}{\quad}$$

$$\vdots$$

$$\frac{}{\to^{L^n} p(v)}\,R^n$$

Consider the sublist $(R^{k_1}, \ldots, R^{k_l})$ of $(R^i)_{i=1..n}$ where $S^{k_i} \backslash S^{k_i-1} \in S$. Let us define $w^i = S^{k_i} \backslash S^{k_i-1}$ and $\widetilde{S^i}$ as $S \backslash \{w_1, \ldots, w^i\}$.

Let us define inductively, for $i = 1..l$ the rules $\widetilde{R^{k_i}}$, $\widetilde{Rm^{k_i}}$ and $\widetilde{L^{k_i}}$. $\widetilde{L^0} = \emptyset$. If

$$R^{k_i}\,\frac{{}_{\Delta_1^i}{\to}^{L_1^i}_{\Gamma_1^i}\ v^i \quad S^{k_i+1}, u_{j_i}(v^i) \to^{L^{k_i}} p(v)}{S^{k_i+1} \to^{L_1^i, L^{k_i}, (\neg\Gamma_1^i, \Gamma_{j_i} \to \neg\Delta(u_{j_i}(u^i))^c)} p(v)}\ \Gamma_{j_i} \in \Delta_1$$

then

$$\widetilde{Rm^{k_i}}\frac{\Delta_1^i \to^{L_1^i}_{\Gamma_1^i} v^i \quad \widetilde{S^i}, u_{j_i}(v^i) \to^{L_1, \widetilde{L^{k_{i-1}}}, \widetilde{L^{i-1}}} p(v)}{\widetilde{S^{i+1}} \to^{L_1^i, L_1, \widetilde{L^{k_{i-1}}}, (\neg \Gamma_1^i, \Gamma_{j_i} \to \neg \Delta(u_{j_i}(u^i))^c)} p(v)} \Gamma_{j_i} \in \Delta_1$$

and $\widetilde{L^{k_i}} = \widetilde{L^{k_{i-1}}}, (\neg \Gamma_1^i, \Gamma_{j_i} \to \neg \Delta(u_{j_i}(u^i))^c)$ and

$$\widetilde{R^{k_i}}\frac{S^{k_i} \backslash S \to^{L^{k_i-1}, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_{i-1}}})} p(v)}{S^{k_i+1} \backslash S \to^{L^{k_i}, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_i}})} p(v)}$$

Note that $\widetilde{R^{k_i}}$ is simply the identity rule. We define $\widetilde{Rm^{k_i}}, \widetilde{L^{k_i}}, \widetilde{R^{k_i}}$ the same way if R^{k_i} is an instance of $(2'_p)_i$

If

$$R^{k_i}\frac{\Delta_1^i \to^{L_1^i}_{\Gamma_1^i} v^i \quad S^{k_i+1}, u_{j_i}(v^i) \to^{L^{k_i}} p(v)}{S^{k_i+1} \to^{L_1^i, L^{k_i}} p(v)} \begin{array}{c} \Gamma_{j_i} \in \Delta_1^i, \Gamma_1^i(S^{k_i+1}), \Gamma_1^i(S^{k_i+1}), \\ \forall (\neg \Gamma \to \neg \Delta) \in L_1^i, L^{k_i}. \Gamma_{j_i}, \Gamma_1^i \notin \Delta \end{array}$$

then

$$\widetilde{Rm^{k_i}}\frac{\Delta_1^i \to^{L_1^i}_{\Gamma_1^i} v^i \quad \widetilde{S^i}, u_{j_i}(v^i) \to^{L_1, \widetilde{L^{k_{i-1}}}, \widetilde{L^{i-1}}} p(v)}{\widetilde{S^{i+1}} \to^{L_1^i, L_1, \widetilde{L^{k_{i-1}}}, (\neg \Gamma_1^i, \Gamma_{j_i} \to \neg \Delta(u_{j_i}(u^i))^c)} p(v)} \Gamma_{j_i} \in \Delta_1$$

and $\widetilde{L^{k_i}} = \widetilde{L^{k_{i-1}}}, (\neg \Gamma_1^i, \Gamma_{j_i} \to \neg \Delta(u_{j_i}(u^i))^c)$ and

$$\widetilde{R^{k_i}}\frac{S^{k_i} \backslash S \to^{L^{k_i-1}, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_{i-1}}})} p(v)}{S^{k_i+1} \backslash S \to^{L^{k_i}, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_i}})} p(v)} \begin{array}{c} \Gamma_{j_i}(S^{k_i} \backslash S), \Gamma_1^i((S^{k_i} \backslash S), \\ \forall (\neg \Gamma \to \neg \Delta) \in L^{k_i}, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_i}}). \Gamma_{j_i}, \Gamma_1^i \notin \Delta \end{array}$$

Note that $\widetilde{R^{k_i}}$ is a valid Remove rule. We define $\widetilde{Rm^{k_i}}, \widetilde{L^{k_i}}, \widetilde{R^{k_i}}$ the same way if R^{k_i} is an instance of $\textsc{Str}_i^1$.

If $j \notin \{k_1, \dots, k_l\}$, if $k_i < j < k_{i+1}$ we define $\widetilde{R^j}$ as

$$\widetilde{R^j}\frac{S^{j-1} \backslash S \to^{L^{j-1}, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_i}})} p(v)}{S^j \backslash S \to^{L^j, (\widetilde{L^{k_l}} \backslash \widetilde{L^{k_i}})} p(v)} R^j$$

Now note that

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{S\ {}_{\Delta_1}\!\rightarrow^{L_1}_{\Gamma_1}\ u}{\vdots}\ \widetilde{R}m^{k_1}}{{}_{\Delta_1}\!\rightarrow^{L_1,\widetilde{L^{k_l}}}_{\Gamma_1}}\ \widetilde{R}m^{k_l} \qquad S', u_i(u) \rightarrow^{L_2} p(v)}{S^0\backslash S \rightarrow^{L^0,\widetilde{L^{k_l}}} p(v)}\ \widetilde{R^0}}{S^1\backslash S \rightarrow^{L^{1'}} p(v)}\ \widetilde{R^1}}{\vdots}}{\rightarrow^{L^n} p(v)}\ \widetilde{R^n}$$

is a valid proof of $\rightarrow^{L^n} p(v)$ with one less non unit cut, which contradict our hypothesis.

Let us call $\mathcal{K}$ the previous set of inference rules.

Lemma 13. *The problem $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n \Vdash_{\mathcal{K}} S \rightarrow t$ is in PTIME*

Proof. In $S\ {}_{\Delta}\!\rightarrow^{L}_{\Gamma}\ t$ it is easy to see that whether the multiplicity of $x \in L$ is 2 or strictly greater than 2 is not relevant, as if x appears 2 times in L and a Remove can be applied for x (yielding a clause C annotated with L' with x of multiplicity 1 in L'), then it can be applied repeatedly if x appears more than twice in order to yield the same C.

First of all note that we know how to decide the problem $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n \Vdash_{\mathcal{K}} \rightarrow t$ as we can apply a unit strategy, and there are only a bounded number of annotations the decision procedure is in PTIME.

Now in order to decide the entailment problem, let Λ be a new constraint such that for all S, $\neg\Lambda(S)$. Now for every clause in $C_i = S_i \rightarrow t_i$ and every constraint Γ let $S_i^{\Gamma} = S_i\backslash\{u \in S | \neg\Gamma(u)\}$ and $C_i^{\Gamma} = S_i^{\Gamma} \rightarrow^{(\neg\Lambda \rightarrow \neg\Gamma)}$. We observe that $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n \Vdash_{\mathcal{K}} S \rightarrow t$ iff there exists Γ such that $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n \Vdash_{\mathcal{K}} \rightarrow^{(\neg\Lambda \rightarrow \neg\Gamma)} t$. Clearly if $S_1 \rightarrow t_1, \ldots, S_n \rightarrow t_n \Vdash_{\mathcal{K}} \rightarrow^{(\neg\Lambda \rightarrow \neg\Gamma)} t$ (le Π be a proof of $\rightarrow^{(\neg\Lambda \rightarrow \neg\Gamma)} t$ in $\mathcal{K}$) then as the $(\neg\Lambda \rightarrow \neg\Gamma')$ annotations can never be removed, replacing the leafs of Π that are C_i^{Γ} by $C_i = C_i^{\Gamma} \cup \{u \in S | \neg\Gamma(u)\}$ yields a proof of $S' \rightarrow t$ with $S' \subseteq S$. Conversely, if there exists a proof of $S' \rightarrow t$ with $S' \subseteq S$ with $S' \subseteq S$ without weakening, then one can build a proof of $\rightarrow^{(\neg\Lambda \rightarrow \neg\Gamma)} t$ from the C_i^{Γ} by backtracking the origin of the atoms in S'.

Theorem 3. *If $\mathcal{S}$ is a set of ground clauses built on $\vdash$, we can decide in PTIME the satisfiability of $\mathcal{S}$ together with T, W and finitely many constrained clauses c_n, c_s, c_c built on the same pattern p, provided the constraints are monotone.*

Proof. Computing the least fixed point of the function defined in the proof of lemma 3 using the oracle computing whether $S_1 \vdash t_1, \ldots, S_n \vdash t_n \vDash_{\mathcal{K}} S \vdash t$ yields a PTIME decision procedure.