Sergiu Bursuc, Hubert Comon-Lundh
and Stéphanie Delaune

Deducibility constraints

Research Report LSV-09-17

July 2009

Laboratoire
Spécification
et
Vérification

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

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

Deducibility constraints *

Sergiu Bursuc[1], Hubert Comon-Lundh[1,2], and Stéphanie Delaune[1]

[1] LSV, CNRS & ENS Cachan & INRIA project SECSI
[2] AIST, Tokyo

Abstract. In their work on tractable deduction systems, D. McAllester and later D. Basin and H. Ganzinger have identified a property of inference systems (the locality property) that ensures the tractability of the Entscheidungsproblem.
On the other hand, deducibility constraints are sequences of deduction problems in which some parts (formulas) are unknown. The problem is to decide their satisfiability and to represent the set of all possible solutions. Such constraints have also been used for deciding some security properties of cryptographic protocols.
In this paper we show that local inference systems (actually a slight modification of such systems) yield not only a tractable deduction problem, but also decidable deducibility constraints. Our algorithm not only allows to decide the existence of a solution, but also gives a representation of all solutions.

1 Introduction

Deciding whether a given statement can be derived from hypotheses, using a set of formal inference rules, is one of the famous issues in proof theory, known as the *Entscheidungsproblem*. This is undecidable for first-order logic and untractable for propositional logic. There are however several formal proof systems for which the problem is tractable, for instance Horn propositional logic, but also the so-called Dolev-Yao intruder deduction rules. D. McAllester [12] observed that *any* inference system which is *local* yields a tractable Entscheidungsproblem. D. Basin and H. Ganzinger [1] proved that locality is equivalent to a saturation property of the set of inference rules.

The Dolev-Yao inference system is local (and saturated with respect to the subterm ordering). This is why deciding whether a message can be computed by an attacker from a finite set of messages can be performed in polynomial (actually linear) time in this formal proof system. Now, if we consider an active attacker, we need not only to solve the Entscheidungsproblem, but a more general problem in which some statements and proofs are unknown: this corresponds to the attacker's choices. This was formalized in [13], using *deducibility constraints*. There are many historical examples of deducibility constraints in mathematics: Fermat gave a proof of his famous theorem, in which parts were missing; filling the holes amounts to solve a deducibility constraint (in formal arithmetic).

* This work has been partially supported by the ANR-07-SESU-002 AVOTÉ

The starting point of the present work is the problem of lifting the results of D. McAllester, D. Basin and H. Ganzinger from deducibility to deducibility constraints. For the security protocols, this corresponds to moving from a passive adversary to an active adversary.

We consider formal inference systems without AC-symbols (as in [1]). We also assume that there is a single unary predicate symbol (this corresponds to the attacker knowledge in security protocols). Then we prove that, for any inference system that is saturated in some suitable way (we call it *good*), the deducibility constraints are decidable. Actually, we prove more: we provide with a constraint simplification algorithm that yields *solved forms*. This allows not only to decide the existence of a solution, but also to represent all solutions. Such a feature is used in [6] for deciding trace properties such as authentication and key cycles in security protocols, and also in [10] for deciding game-theoretic security properties such as abuse-freeness. Our results generalize [6] to any good inference system. Finally, we claim that our transformation rules are simple: we simply guess the last inference step and reflect this on the constraint solving. The difficult part is then the design of a complete and terminating strategy.

As in [1], an advantage of our approach is the ability to complete the original inference system; if the inference system is not good (hence we cannot apply directly our results), we may run a saturation procedure, that will yield a good inference system. Of course, such a procedure may not terminate, in which case we can still use our results on the limit (infinite) inference system, but getting an effective algorithm for solving deducibility constraints requires more work, which is out of the scope of this paper.

Related work. The present work is a strict extension of [6]: we consider a class of inference systems instead of a particular one. The intruder theories that are described by a subterm convergent rewriting system can also be casted as good inference systems (but the converse is false). Hence, as far as trace properties are concerned, we also generalize [2, 3]. Note however that [2] also considers equivalence properties, that are not covered (yet) by our work. There are also several examples of formal (intruder) proof systems that yield decidable deducibility constraints [15, 7, 5, 4, 9]. All these works consider AC-symbols and are incomparable with our results.

Structure of the paper. In Section 2, we introduce good inference systems and their properties. Then, in Section 3, we introduce deducibility constraints. In Section 4, we provide with a set of constraint transformation rules, that is parametrized by any good inference system and that we prove both sound and complete: the solutions of the constraint are the same as the solutions of the solved forms that are obtained by applying the transformation rules. In Section 5, we give a complete and terminating strategy.

2 Preliminaries

In what follows, we assume that $\mathcal{F}$ is a (ranked) alphabet of *function symbols*. *Terms* are built on this set of function symbols and a set of variables $\mathcal{X}$. *Ground* *terms* are terms without variables. For any term t (and, by extension, set of terms or any formal expression), $var(t)$ is the set of variable symbols occurring in t and $st(t)$ denotes the set of *subterms* of t defined as usual.

2.1 Inference systems

We use a natural deduction style for inference systems. An *inference rule* consists in a finite set of terms $\{u_1, \ldots, u_n\}$, the *premises*, and a term u, the *conclusion* such that $var(u) \subseteq var(\{u_1, \ldots, u_n\})$. It is displayed

$$\frac{u_1 \quad \cdots \quad u_n}{u}$$

It may also be convenient to use a deducibility predicate symbol I, in which case the inference rules are simply Horn clauses $I(u_1), \ldots, I(u_n) \rightarrow I(u)$.

Example 1. Consider the signature $\mathcal{F} = \{\mathtt{enc}/3, \mathtt{pub}/1, \mathtt{priv}/1, \langle\, , \,\rangle/2\}$. The symbols $\mathtt{enc}$ and $\langle\, , \,\rangle$ represent respectively probabilistic encryption and pairing, $\mathtt{pub}$ (resp. $\mathtt{priv}$) represents the public key (resp. private key) construction. A possible set of "Dolev-Yao" inference rules for public-key encryption is:

$$(\mathsf{E})\ \frac{x \quad y \quad z}{\mathtt{enc}(x, y, z)} \qquad (\mathsf{D})\ \frac{\mathtt{enc}(\mathtt{pub}(y), x, z) \quad \mathtt{priv}(y)}{x} \qquad (\mathsf{K})\ \frac{x}{\mathtt{pub}(x)}$$

$$(\mathsf{P})\ \frac{x \quad y}{\langle x, y\rangle} \qquad (\mathsf{Proj}_1)\ \frac{\langle x, y\rangle}{x} \qquad (\mathsf{Proj}_2)\ \frac{\langle x, y\rangle}{y}$$

Other relevant examples of inference systems are obtained by adding signature schemes, hash functions, symmetric encryption ...

A *proof*, with hypotheses H and conclusion t is a tree, whose nodes are labeled with terms and such that, if a node is labeled s and its sons are labeled $s_1, \ldots, s_n$, then either $n = 0$ (this is a leaf node), and $s \in H$, or else there is an inference rule whose premises are $u_1, \ldots, u_n$ and conclusion is u and a substitution θ such that $u\theta = s$ and, for every i, $u_i\theta = s_i$. We write $H \vdash t$ when there exists a proof with hypotheses H and conclusion t.

We let $\mathtt{step}(\pi)$ be the set of terms labeling the proof π and $\mathtt{leaves}(\pi)$ be the multiset of the terms that labels the leaves of π. If π is a proof, we let $\mathtt{last}(\pi)$ be the last inference step in π, $\mathtt{premises}(\pi)$ be the proofs of the premises of $\mathtt{last}(\pi)$ and $\mathtt{conc}(\pi)$ be its conclusion. More formally,

$$\text{if} \quad \pi = \frac{\pi_1 \quad \cdots \quad \pi_n}{u} \quad \text{then} \quad \begin{cases} \mathtt{last}(\pi) = \dfrac{\mathtt{conc}(\pi_1) \quad \cdots \mathtt{conc}(\pi_n)}{u} \\ \mathtt{premises}(\pi) = \{\pi_1, \ldots, \pi_n\} \\ \mathtt{conc}(\pi) = u \end{cases}$$

Example 2. Consider the following proof tree π. Actually, π is a proof in the Dolev-Yao inference system presented in Example 1.

$$\dfrac{\mathtt{enc}(\mathtt{pub}(k), a, r) \quad \dfrac{\langle \mathtt{priv}(k), a\rangle}{\mathtt{priv}(k)}\,(\mathsf{Proj}_1)}{a}\,(\mathsf{D})$$

We have that $\mathtt{premises}(\pi) = \{\mathtt{enc}(\mathtt{pub}(k), a, r), \dfrac{\langle \mathtt{priv}(k), a\rangle}{\mathtt{priv}(k)}\}$, $\mathtt{conc}(\pi) = a$, $\mathtt{last}(\pi) = \dfrac{\mathtt{enc}(\mathtt{pub}(k), a, r)\ \mathtt{priv}(k)}{a}$, $\mathtt{leaves}(\pi) = \{\mathtt{enc}(\mathtt{pub}(k), a, r), \langle \mathtt{priv}(k), a\rangle\}$.

2.2 Good inference systems

In the following definitions, we introduce our notion of saturation. Informally, if there is a proof such that some intermediate step is too large (we call this a bad proof and the large step is called a bad pattern), then there must be a simpler proof of the same statement.

If $\mathsf{R} = \dfrac{s_1 \quad \cdots \quad s_n}{s_0}$ is an inference rule, we let $\mathtt{Max}(\mathsf{R})$ be the multiset of the maximal terms s_i, w.r.t. the subterm ordering $\lhd$.

Definition 1 (bad proof / pattern). *A* bad proof *is a proof π of the form:*

$$\dfrac{u_1 \quad \cdots \quad u_n \quad \dfrac{v_1 \quad \cdots \quad v_m}{u_{n+1}}\,\mathsf{R}_1 \quad u_{n+2} \quad \cdots \quad u_{n+k}}{v}\,\mathsf{R}_2$$

such that $\mathsf{R}_1 = \dfrac{s_1 \quad \cdots \quad s_m}{s}$, $\mathsf{R}_2 = \dfrac{t_1 \quad \cdots \quad t_{n+k}}{t}$, $s \in \mathtt{Max}(\mathsf{R}_1)$ *and* $t_{n+1} \in \mathtt{Max}(\mathsf{R}_2)$. *A* bad pattern *in a proof π is a subproof of π of the form:*

$$\dfrac{\pi_2^1 \quad \cdots \quad \pi_2^{i-1} \quad \dfrac{\pi_1^1 \quad \cdots \quad \pi_1^m}{\mathtt{conc}(\pi_2^i)}\,\mathsf{R}_1 \quad \pi_2^{i+1} \quad \cdots \quad \pi_2^n}{v}\,\mathsf{R}_2$$

such that the following proof is a bad proof.

$$\dfrac{\mathtt{conc}(\pi_2^1) \ \cdots \ \mathtt{conc}(\pi_2^{i-1}) \quad \dfrac{\mathtt{conc}(\pi_1^1) \ \cdots \ \mathtt{conc}(\pi_1^m)}{\mathtt{conc}(\pi_2^i)} \quad \mathtt{conc}(\pi_2^{i+1}) \ \cdots \ \mathtt{conc}(\pi_2^n)}{v}$$

If $\pi = \mathsf{R}\theta$ is an instance of an inference rule R and $\mathtt{Max}(\mathsf{R}) = \{s_1, \ldots, s_k\}$, then $\mu(\pi)$ is the multiset $\{s_1\theta, \ldots, s_k\theta\}$. If π is a proof, $\mu(\pi)$ is defined as the multiset of $\mu(\pi')$ for all inference steps π' of π. Formally, if $\mathtt{premises}(\pi) = \{\pi_1, \ldots, \pi_n\}$, we have that:

$$\mu(\pi) = \mu(\pi_1) \uplus \cdots \uplus \mu(\pi_n) \uplus \mu(\mathtt{last}(\pi)).$$

Multisets are ordered using the multiset extensions of their elements: if $\succeq$ is an ordering, we let $\succeq_m$ be its multiset extension.

Definition 2 (good inference system). *An inference system is* good *if there is a total well-founded extension $\prec$ of the subterm ordering $\lhd$ such that, for any bad proof π, there is a proof π' of $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$ with $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$ (multiset inclusion), $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$, and $\mu(\pi') \neq \mu(\pi)$.*

Example 3. The Dolev-Yao inference system described in Example 1 is good. Indeed, all bad proofs are of one of the following forms

$$\dfrac{\dfrac{\mathtt{pub}(u_1)\quad u_2\quad u_3}{\mathtt{enc}(\mathtt{pub}(u_1), u_2, u_3)}\quad \mathtt{priv}(u_1)}{u_2} \qquad\qquad \dfrac{\dfrac{u_1 \quad u_2}{\langle u_1, u_2\rangle}}{u_i}\quad i = 1, 2$$

Obviously, for all such π there is a smaller, trivial, proof π' of $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$ such that $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$ and $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$ for any total well-founded extension $\prec$ of the subterm ordering.

Now, we give another example in which the inference system is no longer finite. We consider blind signatures, as described in [8], that are used in some e-voting protocols. The inference system is not good. However, we may complete it and get an infinite, yet recursive, good inference system.

Example 4. We add the following rules to the system of Example 1:

$$(\mathsf{S})\ \dfrac{x \quad y}{\mathtt{sign}(x, y)} \qquad (\mathsf{C})\ \dfrac{\mathtt{sign}(x, y)}{x}$$

$$(\mathsf{B})\ \dfrac{x \quad y}{\mathtt{blind}(x, y)} \qquad (\mathsf{UB}_1)\ \dfrac{\mathtt{blind}(x, y)\quad y}{x} \qquad (\mathsf{UB}_2)\ \dfrac{\mathtt{sign}(\mathtt{blind}(x, y), z)\quad y}{\mathtt{sign}(x, z)}$$

Because of the rule (UB_2), the system not good. The following proof π is bad:

$$\dfrac{\dfrac{\mathtt{sign}(\mathtt{blind}(\mathtt{blind}(x, x_1), x_2), y)\quad x_2}{\mathtt{sign}(\mathtt{blind}(x, x_1), y)}\qquad x_1}{\mathtt{sign}(x, y)}$$

There is no other proof π' of $\mathtt{leaves}(\pi) \vdash \mathtt{sign}(x, y)$ such that $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. Thus, for any total well-founded extension $\prec$ of $\lhd$, there is no proof π' of $\mathtt{leaves}(\pi) \vdash \mathtt{sign}(x, y)$ such that $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$ and $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$. However, we may add all shortcuts that correspond to bad proofs. Let $b_n(x, x_1, \ldots, x_n)$ be defined by $b_1(x, x_1) = \mathtt{blind}(x, x_1)$ and $b_{n+1}(x, x_1, \ldots, x_{n+1}) = \mathtt{blind}(b_n(x, x_1, \ldots, x_n), x_{n+1})$. We add the following rules (for every $n \geq 1$) and the resulting system is a good inference system (see Appendix A.1).

$$\dfrac{\mathtt{sign}(b_n(x, x_1, \ldots, x_n), y)\quad x_1\ \ldots\ x_n}{\mathtt{sign}(x, y)}$$

2.3 Some properties of good inference systems

Definition 3 (simple proof). *Let $H_1 \subseteq H_2 \subseteq \cdots \subseteq H_n$. A proof π of $H_i \vdash u$ is* left-minimal *if for any $j < i$ such that $H_j \vdash u$, π is a proof of $H_j \vdash u$. A proof is* simple *if it does not contain any bad pattern and all its subproofs are left-minimal.*

Example 5. Consider the Dolev-Yao inference system given in Example 1. Let $H_1 = \{\mathtt{enc}(\mathtt{pub}(k), a, r), \mathtt{priv}(k), a\}$, $H_2 = H_1 \cup \{\langle a, b\rangle\}$. We have that $H_2 \vdash a$. Indeed, the proofs π_1, π_2 and π_3 described below are witnesses of this fact:

$$\frac{\langle a, b\rangle}{a} \qquad \frac{\mathtt{enc}(\mathtt{pub}(k), a, r) \quad \mathtt{priv}(k)}{a} \qquad \frac{\dfrac{a \quad a}{\langle a, a\rangle}}{a}$$

The proofs π_2 is simple whereas π_1 and π_3 are not. Note that proof of $H_2 \vdash a$ reduced to a leaf is also a simple proof.

Lemma 1. *Consider a good inference system. Let $H_1 \subseteq H_2 \subseteq \cdots \subseteq H_n$ be an increasing sequence of sets of terms and $i \in \{1, \ldots, n\}$. If π is a proof of $H_i \vdash u$ then there is a simple proof of $H_i \vdash u$.*

From now on, we only consider good inference systems. The rules of such systems can be divided in three sets. The *composition rules* whose conclusion is the only maximal term. Any rule $I(x_1), \ldots, I(x_n) \rightarrow I(\mathsf{f}(x_1, \ldots, x_n))$ is a composition, e.g. (P), (E), (S). The *decomposition rules* whose all maximal terms are premises, e.g. (D), (UB_1). The *versatile rules* whose both the conclusion and some premises are maximal, e.g. (UB_2). In what follows, we also assume that:

1. any composition rule has a conclusion $\mathsf{f}(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are variables. This is the case in our application area: each function symbol is either public (and there is such a rule) or private.
2. any versatile rule satisfies the following properties:
   (a) each strict subterm of the conclusion is a subterm of some premise.
   (b) each premise that is not maximal in the rule is a strict subterm of another premise of that rule.

These conditions are satisfied in Examples 1 and 4. Besides these examples, any intruder theory that can be presented by a finite subterm-convergent rewrite system satisfies our hypotheses (see Appendix A.2). These hypotheses might not be necessary for our result, but we use them in our proof.

We now classify the proofs, according to the type of the last proof step. This generalizes the classical composition/decomposition classification:

Lemma 2 (locality). *Let π be a proof of $H \vdash u$ without bad pattern, one of the following occurs:*

- $\mathtt{last}(\pi)$ *is a composition and* $\mathtt{step}(\pi) \subseteq st(H \cup \{u\})$*;*
- π *is reduced to a leaf or* $\mathtt{last}(\pi)$ *is a decomposition and* $\mathtt{step}(\pi) \subseteq st(H)$*;*
- $\mathtt{last}(\pi)$ *is (an instance of) a versatile rule and* $\mathtt{step}(\pi') \subseteq st(H)$ *for any strict subproof* π' *of* π*.*

This is proved by observing that any proof in which a maximal conclusion is also a maximal premise of the next rule can be simplified, according to the definition of good inference systems.

3 Deducibility constraints

The following definition of (deducibility) constraints has been proved to be relevant in the context of security protocols verification (see, e.g. [14, 15, 6]).

Definition 4. *A* constraint system D *is a formula of the form* $\exists\tilde{z}.[C \mid \mathsf{E}]$ *where:*

- $\tilde{z}$ *is a sequence of variables;*
- $\mathsf{E}(D) = \mathsf{E}$ *is a set of equations in solved form, identified to a substitution* θ_{E}*;*
- C *is a conjunction of deducibility constraints* $H_1 \Vdash u_1 \wedge \ldots \wedge H_n \Vdash u_n$ *where* $var(C) \cap dom(\theta_{\mathsf{E}}) = \emptyset$*,* $H_1, \ldots, H_n$ *are finite sets of terms,* $u_1, \ldots, u_n$ *are terms, and such that* monotony *and* origination *are satisfied:*
  - Monotony*:* $\emptyset \neq H_1 \subseteq H_2 \subseteq \ldots \subseteq H_n$*;*
  - Origination*:* $var(H_i) \subseteq var(\{u_j \mid H_j \subsetneq H_i\})$ *for* $1 \leq i \leq n$*.*

We let $fvar(D) = var(D) \smallsetminus \tilde{z}$ *and* $\mathsf{LH}(D) = \{H_1, \ldots, H_n\}$*.*

Definition 5 (solution). *Given an inference system,* a solution of *a constraint system* $D = \exists\tilde{z}.[C \mid \mathsf{E}]$ *is a ground substitution* σ *with* $dom(\sigma) = fvar(D)$ *such that there is a ground substitution* τ *with* $dom(\tau) = \tilde{z}$ *such that:*

- $H(\sigma \cup \tau) \vdash u(\sigma \cup \tau)$ *for every* $H \Vdash u \in C$*, and*
- $u(\sigma \cup \tau) = v(\sigma \cup \tau)$ *for every* $u = v \in \mathsf{E}$*.*

We let $Sol(D)$ *be the set of solutions of* D*.*

In the context of security protocols, any solution will correspond to a choice of messages that are constructed by the attacker and that are accepted by the honest parties.

Example 6. We consider the Dolev-Yao inference system given in Example 1.

$$D := \begin{cases} H_1 = a \Vdash x_0 \ \wedge \ a \Vdash x_1 \\ H_2 = \mathtt{enc}(x_0, \langle b, x_1 \rangle, r),\ \mathtt{priv}(a),\ a \Vdash b \end{cases}$$

$H_1 \subseteq H_2$ and the variables x_0, x_1 occur first on the right. Thus, D is a constraint system. $\sigma = \{x_0 \mapsto \mathtt{pub}(a), x_1 \mapsto \langle a, a \rangle\}$ is a solution of D. Here, there are no bounded variables nor equations. This is the case for constraint systems that represent the security protocol executions. Bounded variables and equations may however be introduced by our constraint solving rules.

Putting together Definitions 4 and 5, we get the following problem, whose decision is the subject of this paper:

> Given an inference system and a constraint system D, does there exist a substitution σ such that $\sigma \in Sol(D)$? We also want to find an effective representation of all solutions.

Notation. Let $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ be a constraint system. For every variable $x \in var(D)$, we let H_x be the smallest set $H \in \mathsf{LH}(D)$ such that there is a constraint $H \Vdash u \in D$ with $x \in var(u) \smallsetminus var(H)$. In other words, H_x is the left hand side of the deducibility constraint that introduced the variable x for the first time. By origination and monotony, this is defined for all $x \in var(C)$. By convention, $H_x = \emptyset$ when x does not occur in C.

4 Transformation of deducibility constraints

We show here that we can solve deducibility constraints in such a way that *we do not miss any solution* (as in [6]). The basic idea of the transformation rules is very straightforward, and that is what makes it appealing: we simply guess the last step of the proof, performing a backwards proof search together with narrowing the variables of the constraint. If $\mathsf{R} = I(u_1), \ldots, I(u_n) \rightarrow I(u)$ is guessed as the last rule in the proof of $H\sigma \vdash v\sigma$, we simply perform:

$$\exists \tilde{z}.[C \wedge H \Vdash v \mid \mathsf{E}] \rightsquigarrow \exists \tilde{z'}.[C\theta \wedge H\theta \Vdash u_1\theta \wedge \ldots \wedge H\theta \Vdash u_n\theta \mid \mathsf{E}']$$

where $\tilde{z'} = \tilde{z} \cup var(\mathsf{R})$, $\theta = \mathsf{mgu}(u, v)$ and $\mathsf{E}' = \mathsf{E} \cup \theta$.

This hardly terminates, even for very simple proof systems and ground goals. Consider the rule (Proj_1) only. We get:

$$\begin{aligned} H \Vdash v &\rightsquigarrow \exists x_1, x_2.[H \Vdash \langle v, x_2\rangle \mid x_1 = v] \\ &\rightsquigarrow \exists x_1, x_2, y_1, y_2.[H \Vdash \langle\langle v, x_2\rangle, y_2\rangle \mid x_1 = v \wedge y_1 = \langle v, x_2\rangle] \rightsquigarrow \ldots \end{aligned}$$

And similarly for (P) (below, we assume that H is a ground set of terms):

$$H \Vdash x \rightsquigarrow \exists x_1, x_2.[H \Vdash x_1 \wedge H \Vdash x_2 \mid x = \langle x_1, x_2\rangle] \rightsquigarrow \ldots$$

First, we do not aim at explictly enumerating all possible solutions, but only compute *solved forms*, that are a convenient representation of all these solutions. Typically, $H \Vdash v$ will be solved when v is a variable. This rules out the second above non-terminating example.

For decomposition or versatile rules, we may still get the first non-terminating behavior. That is where we use locality: we control the application of such rules, roughly requesting that maximal premises are subterms of H. This is however not complete, as Lemma 2 shows only that, in case of a versatile or decomposition rule, the premises are subterms of the hypotheses *at the ground level.* In other words, if we guessed that the last rule, in the proof of an instance σ of $H \Vdash v$, is a decomposition, we only know that the premises of the last proof step are in $st(H\sigma)$. We use then the property of subterms: $st(H\sigma) = st(H)\sigma \cup st(\sigma)$. If the premises are in $st(H)\sigma$, everything is fine: we can guess subterms of H that are the premises. Otherwise, it is not so straightforward, as σ is unknown. That is where we need some additional strategies.

Another difficulty comes from the introduction of variables. If we keep on introducing variables and equations, the left hand sides of deducibility constraints may grow, hence their subterms too. Then guessing a subterm of H as a premise does not necessarily yield a bounded number of terms.

4.1 Transformation rules

The rules of Figure 1 are applied non-deterministically. When new variables are introduced (in the Dec rule) they are assumed to be fresh, by renaming.

(Axiom) $\exists\tilde{z}.[C \wedge H \Vdash u \mid \sigma] \rightsquigarrow \exists\tilde{z}.[C\theta \mid \sigma \cup \theta]$
where $\theta = \mathsf{mgu}(u, v)$, $v \in H$ and $u \notin \mathcal{X}$

(Triv) $\exists\tilde{z}.[C \wedge H \Vdash x \wedge H' \Vdash x \mid \sigma] \rightsquigarrow \exists\tilde{z}.[C \wedge H \Vdash x \mid \sigma]$
when $H \subseteq H'$

(Comp) $\exists\tilde{z}.[C \wedge H \Vdash \mathsf{f}(u_1, \ldots, u_n) \mid \sigma] \rightsquigarrow \exists\tilde{z}.[C \wedge H \Vdash u_1 \wedge \ldots \wedge H \Vdash u_n \mid \sigma]$
if f is a public symbol

(Dec) $\exists\tilde{z}.[C \wedge H \Vdash v \mid \sigma] \rightsquigarrow \exists\tilde{z} \cup \tilde{x}.[C\theta \wedge H\theta \Vdash w_1\theta \wedge \ldots H\theta \Vdash w_n\theta \quad \mid \sigma \cup \theta]$
$\wedge H'\theta \Vdash v_1\theta \wedge \ldots \wedge H'\theta \Vdash v_m\theta$

where:

- $\mathsf{R} = \dfrac{v_1 \quad \ldots \quad v_m \quad w_1 \quad \ldots \quad w_n}{w}$ is a decomposition or a versatile rule such that $\mathtt{Max}(\mathsf{R}) \subseteq \{w_1, \ldots, w_n\}$ and $\tilde{x} = var(\mathsf{R})$;
- $\theta = \mathsf{mgu}(\langle w, w_1, \ldots, w_n\rangle, \langle v, u_1, \ldots, u_n\rangle)$, $u_1, \ldots, u_n \in st(H) \smallsetminus \mathcal{X}$, and $v \notin \mathcal{X}$;
- H' is a left member of a deducibility constraint such that $H' \subsetneq H$.

Fig. 1. Transformation of deducibility constraints

The Dec rule deserves some explanation. We guessed here a versatile or decomposition rule. The premises $w_1, \ldots, w_n$ will be those whose instances correspond to a term in $st(H)\sigma$: we can guess the corresponding terms in $st(H)$, namely $u_1, \ldots, u_n$. The other premises (that are then subterms in the substitution part) are constrained to be proved with strictly less hypotheses. We will show that this can always be assumed, hence that we get completeness.

Example 7. Consider the constraint system D given in Example 6. First, considering the rule (Proj_1) and applying Dec to the third constraint yields:

$\exists x', y'.[a \Vdash x_0, a \Vdash x_1, \mathtt{enc}(x_0, \langle b, x_1\rangle, r), \mathtt{priv}(a), a \Vdash \langle b, x_1\rangle \mid \{x' \mapsto b, y' \mapsto x_1\}]$.

Now, considering (D) and applying again Dec to the third constraint yields:

$$D' = \begin{cases} \exists x, y, z, x', y'.[\, a \Vdash x_0\theta, \quad a \Vdash x_1\theta \\ \qquad\qquad\qquad H_2\theta \Vdash \mathtt{enc}(x_0\theta, \langle b, x_1\theta\rangle, r) \\ \qquad\qquad\qquad H_2\theta \Vdash \mathtt{priv}(a) \ \mid \ \theta \cup \{x' \mapsto b, y' \mapsto x_1\}] \end{cases}$$

where $\theta = \mathsf{mgu}(\langle x, \mathtt{enc}(\mathtt{pub}(y), x, z), \mathtt{priv}(y)\rangle, \langle\langle b, x_1\rangle, \mathtt{enc}(x_0, \langle b, x_1\rangle, r), \mathtt{priv}(a)\rangle)$
$= \{x \mapsto \langle b, x_1\rangle,\ y \mapsto a,\ z \mapsto r,\ x_0 \mapsto \mathtt{pub}(a)\}$.

Lemma 3 (soundness). *Let D be a constraint system such that $D \rightsquigarrow D'$, then D' is a constraint system and $Sol(D') \subseteq Sol(D)$.*

The transformation rules of Figure 1 also preserve the following invariant.

Definition 6 (uniquely determined). *Let $D = \exists\tilde{z}.[C \mid \mathsf{E}]$ be a constraint system. D is* uniquely determined *if for any ground substitution σ such that $dom(\sigma) = fvar(D)$, there are ground terms $u_1, \ldots, u_\ell$ such that either $\mathsf{mgu}(\mathsf{E}\sigma) = \perp$ or $\mathsf{mgu}(\mathsf{E}\sigma) = \{z_1 = u_1, \ldots, z_\ell = u_\ell\}$ where $\tilde{z} = \{z_1, \ldots, z_\ell\}$. In that case we let $\overline{\sigma}$ be $\sigma \cup \mathsf{mgu}(\mathsf{E}\sigma)$. ($\overline{\sigma}$ is a solution of the constraint system $[C \mid \mathsf{E}]$.)*

Example 8. Let D' be the constraint system given in Example 7. We have that $fvar(D) = \{x_0, x_1\}$. Once values are assigned to x_0, x_1, there is a unique substitution τ that satisfies the equations in $\mathsf{E}(D')$.

Lemma 4. *Let D be a constraint system that is uniquely determined and D' be such that $D \rightsquigarrow D'$. Then D' is uniquely determined.*

Using our transformation rules, solving deducibility constraint systems can be reduced to solving simpler constraint systems that we call solved.

Definition 7. *A constraint system $D = \exists\tilde{z}.[H_1 \Vdash x_1 \wedge \ldots \wedge H_n \Vdash x_n \mid \mathsf{E}]$ is in* solved form *when $x_1, \ldots, x_n$ are distinct variables.*

Solved deducibility constraint systems are particulary simple since they always have a solution.

Lemma 5. *A solved form has always at least one solution.*

4.2 Completeness

Let $H_1 \subseteq H_2 \subseteq \ldots \subset H_n$be a sequence of sets of terms. Let π be a proof of $H_i \vdash u$ for some i ($1 \leq i \leq n$). We associate to π, the minimal set $\mathtt{Hyp}(\pi) \in \{H_1, \ldots, H_n\}$ containing the leaves of π. Note that $\mathtt{Hyp}(\pi) \subseteq H_i$. Given a constraint system $D = \exists\tilde{z}.[C \mid \mathsf{E}]$ that is uniquely determined and a solution σ, a simple proof w.r.t. D is a simple proof w.r.t. the sequence of sets of terms $\mathsf{LH}(D)\overline{\sigma}$.
We first show that either the subterms occurring in proofs are subterms of the hypotheses, or else their simple proofs end with a composition or a versatile rule.

Lemma 6. *Let D be a constraint system of the form $[C \mid \mathsf{E}]$. Let $H \in \mathsf{LH}(D)$ be such that for every $y \in var(H)$ there is a constraint $H_y \Vdash y \in D$. Let σ be a solution of D and v be a term such that $H\sigma \vdash v$. Let $u \in st(v)$. Then:*

1. *either $u \in (st(H) \smallsetminus \mathcal{X})\sigma$;*
2. *or $H\sigma \vdash u$ and any simple proof π of $H\sigma \vdash u$ ends with a composition or a versatile rule.*

To prove this lemma, we consider the set Π of simple proofs of $H\sigma \vdash v$ and we prove the lemma by induction on the pair (H, d) where d is the size of a minimal proof in Π (see Appendix B).

Now, we define the complexity of the proofs witnessing that σ is a solution of D and show that there is always a rule yielding a strictly smaller complexity, until we reach a solved form. Let $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ be a uniquely determined constraint system and σ be a solution of D.

- If $H \Vdash u \in C$ then $\mathsf{PS}(H \Vdash u, \sigma)$ is the size (i.e. number of nodes) of a simple proof of $H\overline{\sigma} \vdash u\overline{\sigma}$ that has a minimal size.
- If $\mathsf{LH}(D) = \{H_1, \ldots, H_n\}$ with $H_1 \subsetneq \ldots \subsetneq H_n$, then the *level* $\mathsf{lev}(H \Vdash u, D)$ of a deducibility constraint $H \Vdash u \in C$ is the index i such that $H = H_i$.

The measure PS is extended to constraint systems by letting, for any solution σ of D, $\mathsf{PS}(D, \sigma)$ be the multiset of pairs $(\mathsf{lev}(H \Vdash u, D), \mathsf{PS}(H \Vdash u, \sigma))$ for all deducibility constraints $H \Vdash u \in D$. The multisets $\mathsf{PS}(D, \sigma)$ are compared using the multiset extension of the lexicographic composition of the orderings.

Note that the number of different levels in a constraint system might decrease, but it may never increase.

Lemma 7. *If D is a constraint system that is uniquely determined and $\sigma \in Sol(D)$, then either D is in solved form or else there is a D' such that $D \rightsquigarrow D'$, $\sigma \in Sol(D')$, and $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$.*

Proof. (sketch) If D is not in solved form, there must be a constraint $H \Vdash u \in D$ such that u is not a variable. We consider such a constraint, with a minimal left hand side. Then, depending on the last rule of a minimal size simple proof of $H\sigma \vdash u\sigma$, we may apply some transformation, that yields a smaller PS:

- *If the proof is reduced to a leaf*, then we use the Axiom rule.
- *If the last rule is a composition*, then we apply Comp to D, yielding a smaller PS.
- *If the last rule is versatile or a decomposition*, we have to show that the conditions of Dec are met in order to conclude. To prove this, we rely on Lemma 2 and Lemma 6. □

Then, we prove the following lemma by induction on $\mathsf{PS}(D, \sigma)$, applying Lemma 7 for the induction step. Lemma 4 allows us to ensure that the resulting constraint system is uniquely determined and to apply our induction hypothesis.

Lemma 8 (completeness). *If D is a constraint system that is uniquely determined and $\sigma \in Sol(D)$, then there is a solved deducibility constraint D' such that $D \rightsquigarrow^* D'$ and $\sigma \in Sol(D')$.*

5 Termination

In order to get termination, we add some control on the transformation rules.

5.1 Our strategy

For every variable x and constraint system D, the *level* $\mathsf{lev}(x, D)$ of x is the level of H_x in D, if $x \in var(D)$, and is 0 otherwise. The deducibility constraints of D are split into an *active part* $\mathsf{Act}(D)$ and a *frozen part* $\mathsf{Fr}(D)$.

Definition 8. *An* extended constraint system D *is a formula* $\exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$ *where:*

- $\tilde{z}$ *is a sequence of variables;*
- $\mathsf{E}(D) \stackrel{\mathsf{def}}{=} \mathsf{E}$ *is a set of equations (not necessarily in solved form) with* $\mathsf{mgu}(\mathsf{E}) \neq \perp$*;*
- $\mathsf{Act}(D) \stackrel{\mathsf{def}}{=} \mathsf{A}$*, the* active part *of* D*, and* $\mathsf{Fr}(D) \stackrel{\mathsf{def}}{=} \mathsf{F}$*, the* frozen part *of* D*, are sets of deducibility constraints;* A *and* $(\mathsf{A} \cup \mathsf{F})\mathsf{mgu}(\mathsf{E})$ *are constraint systems.*

Let $\theta = \mathsf{mgu}(\mathsf{E})$*. A solution of* $\exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$ *is a solution of* $\exists\tilde{z}.[(\mathsf{A} \cup \mathsf{F})\theta \mid \theta]$*. The system* D *is in solved form when* $\mathsf{Fr}(D) = \emptyset$ *and* $\mathsf{Act}(D)\theta$ *is in solved form.*

The rules are described in Figure 2. The transformation relation defined by these rules is denoted $\mapsto$. Sometimes, we use $\mapsto_{\mathsf{A/F}}$ instead of $\mapsto_{\mathsf{A}} \cup \mapsto_{\mathsf{F}}$.

In the initial constraint system, nothing is frozen. All rules only apply to the active part and all rules (except Dec) only modify the active part.

- When the rule Dec is applied to a constraint $H \Vdash v$ such that, for some variable $x \in var(v)$, we have that $\mathsf{lev}(x, \mathsf{Act}(D)) = \mathsf{lev}(H, \mathsf{Act}(D))$, it also contributes only to the active part.
- Otherwise, when for all $x \in var(v)$, $\mathsf{lev}(x, \mathsf{Act}(D)) < \mathsf{lev}(H, \mathsf{Act}(D))$, then only the constraints $H \Vdash u_1 \wedge \ldots \wedge H \Vdash u_n$ are kept in the active part, and the remainder falls in the frozen part.

When $\mathsf{Act}(D)$ is in solved form (and only then), we open the fridge and pour the frozen part into the active one, performing all necessary replacements.

First, we have to establish the soundness of the transformation rules. The main point is to show that the active part remains a constraint system.

Lemma 9 (soundness). *Let D be an extended constraint system such that $D \mapsto D'$ then D' is an extended constraint system and $Sol(D') \subseteq Sol(D)$.*

If there is a loop on the active part by using only Active and Freeze, i.e. $D \mapsto^* D_1 \mapsto^*_{\mathsf{A/F}} D_2$ and $\mathsf{Act}(D_1) = \mathsf{Act}(D_2)$, we remove all the branches that begin with this prefix. We will show that this strategy is both:

(Active) $\exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}] \mapsto_{\mathsf{A}} \exists\tilde{z}\cup\tilde{x}.[\mathsf{A}' \mid \mathsf{F} \mid \mathsf{E}\cup\theta]$

if $\mathsf{A} \rightsquigarrow \exists\tilde{x}.[\mathsf{A}' \mid \theta]$ using Axiom, Triv, Comp; or Dec on $H \Vdash v$ and there exists $x \in var(v)$ such that $\mathsf{lev}(x,\mathsf{A}) = \mathsf{lev}(H,\mathsf{A})$.We assume that $\mathsf{mgu}(\mathsf{E}\cup\theta) \neq \bot$.

(Freeze) $\exists\tilde{z}.[\mathsf{A} \wedge H \Vdash v \mid \mathsf{F} \mid \mathsf{E}] \;\mapsto\; \exists\tilde{z}\cup\tilde{x}.[\mathsf{A} \wedge H \Vdash u_1 \wedge \ldots H \Vdash u_n \mid \mathsf{F} \wedge H' \Vdash v_1 \wedge \ldots \wedge H' \Vdash v_m \mid \mathsf{E}\cup\theta]$

where:

- $\mathsf{R} = \dfrac{v_1 \;\ldots\; v_m \;\; w_1 \;\ldots\; w_n}{w}$ is a decomposition or a versatile rule such that $\mathtt{Max}(\mathsf{R}) \subseteq \{w_1,\ldots,w_n\}$ and $\tilde{x} = var(\mathsf{R})$;
- $\theta = \mathsf{mgu}(\langle w, w_1,\ldots,w_n\rangle, \langle v, u_1,\ldots,u_n\rangle)$, $u_1,\ldots,u_n \in st(H) \smallsetminus \mathcal{X}$, and $v \notin \mathcal{X}$;
- H' is a left member of a deducibility constraint in A such that $H' \subsetneq H$;
- $\mathsf{mgu}(\mathsf{E}\cup\theta) \neq \bot$ and $\mathsf{lev}(x, \mathsf{A}\wedge H \Vdash v) < \mathsf{lev}(H, A \wedge H \Vdash v)$ for any $x \in var(v)$.

(Open) $\exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}] \mapsto_{\mathsf{O}} \exists\tilde{z}.[(\mathsf{A}\cup\mathsf{F})\theta \mid \emptyset \mid \theta]$

when A in solved form and $\theta = \mathsf{mgu}(\mathsf{E})$

Fig. 2. Transformation of extended constraint systems

- *complete*: for any D, for any $\sigma \in Sol(D)$, there is a sequence $D \mapsto^* D'$ authorized by the strategy such that $\sigma \in Sol(D')$ and D' is in solved form.
- *terminating*: there are no infinite transformation sequences.

5.2 Termination of our strategy

Now, we clarify the role of the fridge, by showing that the level of a variable in the active part is never increasing. Moreover, if new variables are introduced in the active part, their level is strictly smaller than the level of an older variable, whose respective level strictly decreased.

Lemma 10. *If $D \mapsto_{\mathsf{A/F}} D'$, then:*

1. $\mathsf{lev}(x, \mathsf{Act}(D')) \leq \mathsf{lev}(x, \mathsf{Act}(D))$ *for every* $x \in var(\mathsf{Act}(D))$.
2. *Let* $M = var(\mathsf{Act}(D')) \smallsetminus var(\mathsf{Act}(D))$. *If* $M \neq \emptyset$ *then there exists* $x \in var(\mathsf{Act}(D))$ *such that* $\mathsf{lev}(z, \mathsf{Act}(D')) < \mathsf{lev}(x, \mathsf{Act}(D))$ *for any* $z \in M \cup \{x\}$.

Now, we get a first termination lemma:

Lemma 11. *There is no infinite transformation sequence without opening the fridge.*

The reason for this is that either we do not introduce variables in which case there must be a loop (this is forbidden by the strategy) or else, thanks to Lemma 10, there is a level ℓ such that the level of some variable at level ℓ strictly decreases, while all new variables have a strictly smaller level than ℓ.

Finally, we cannot open very often the fridge:

Lemma 12. *Let D be an extended constraint system such that $\mathsf{E}(D) = \mathsf{Fr}(D) = \emptyset$. In any transformation sequence starting with D, we can open at most $2|\mathsf{LH}(D)| + |\mathsf{MLH}(D)|$ times the fridge – $\mathsf{MLH}(D)$ is the maximal set of hypotheses in $\mathsf{LH}(D)$.*

The idea is that the maximal level variables x occur in a constraint $H \Vdash x$ when we are about to open the fridge. Furthermore, in the fridge, all variables have a strictly smaller level. It follows that $H \Vdash x$ will not participate any more in any transformation: further transformations only take place at lower levels. More precisely, we show that between two openings, the number of levels or the number of distinct terms in the maximal left hand side decreases; otherwise in the resulting system further transformations only take place at lower levels.

From the previous lemmas, we derive the following corollary.

Corollary 1 (termination). *Our strategy is strongly terminating: there is no infinite transformation sequence of (extended) deducibility constraints.*

5.3 Completeness of our strategy

For an extended constraint system $D = \exists \tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$, we let $\mathsf{Open}(D) = \exists \tilde{z}.[(\mathsf{A} \cup \mathsf{F})\mathsf{mgu}(\mathsf{E}) \mid \mathsf{mgu}(\mathsf{E})]$ and we say that D is uniquely determined if $\mathsf{Open}(D)$ is uniquely determined. Note that by definition of Open, we have that $Sol(D) = Sol(\mathsf{Open}(D))$ and "uniquely determined" is preserved by $\mapsto$.

The following corollary ensures that PS is decreasing on the active part.

Corollary 2 (of Lemma 7). *Let D be a uniquely determined extended constraint system such that $\mathsf{Act}(D)$ is not in solved form and $\sigma \in Sol(D)$. Then there is a D' such that $D \mapsto_{\mathsf{A/F}} D'$ and $\sigma \in Sol(D')$.*

Moreover, $\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{var(\mathsf{A})}) > \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{var(\mathsf{A}')})$ where $\mathsf{A} = \mathsf{Act}(D)$, $\mathsf{A}' = \mathsf{Act}(D')$ and $\overline{\sigma}$ is the extension of σ w.r.t. D'.

Thanks to this, if there is a loop on the active part, we can close the branch, still having a complete strategy:

Lemma 13. *Our strategy is complete.*

6 Conclusion

We gave a simple set of transformation rules that allows to derive a complete and effective representation of all solutions of a deducibility constraint. This works for any good inference system that satisfies some additional syntactic conditions. We believe that this is the starting point of several further works:

1. It would be nice to remove the additional syntactic restrictions (or to prove that they are necessary)
2. Getting a full generalization of [1], requires to introduce predicate symbols.
3. We need to enrich the syntax of constraints, in order to get effective algorithms for infinite (recursive) good inference systems.

4. Our transformation rules are not only preserving all solutions, but also all simple proofs, i.e. some witnesses that they are indeed solutions. This suggests that the same transformation rules can be used for the decision of the symbolic equivalence of constraint systems.
5. Covering all current decision results requires an extension that includes AC-symbols.

References

1. D. Basin and H. Ganzinger. Automated complexity analysis based on ordered resolution. *Journal of the ACM*, 48(1):70–109, 2001.
2. M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proc. 12th ACM Conference on Computer and Communications Security (CCS'05)*, pages 16–25, Alexandria, Virginia, USA, 2005. ACM Press.
3. Y. Chevalier and M. Kourjieh. Key substitution in the symbolic analysis of cryptographic protocols. In *Proc. 27th International Conference on Foundations of Software Technology and Theoretical Computer Science (FTS&TCS'07)*, volume 4855 of *LNCS*. Springer, 2007.
4. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. In *Proc. 23th International Conference on Foundations of Software Technology and Theoretical Computer Science (FTS&TCS'03)*, volume 2914 of *LNCS*, 2003.
5. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with xor. In Kolaitis [11].
6. H. Comon-Lundh, V. Cortier, and E. Zalinescu. Deciding security properties of cryptographic protocols. application to key cycles. *Transaction on Computational Logic*, 2009. To appear.
7. H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in preence of exclusive or. In Kolaitis [11].
8. S. Delaune, S. Kremer, and M. D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.
9. S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis for monoidal equational theories. *Information and Computation*, 206(2-4):312–351, 2008.
10. D. Kähler and R. Küsters. Constraint Solving for Contract-Signing Protocols. In *Proc. 16th International Conference on Concurrency Theory (CONCUR 2005)*, volume 3653 of *LNCS*, pages 233–247. Springer, 2005.
11. P. Kolaitis, editor. *18th Annual IEEE Symposium on Logic in Computer Science.* IEEE Comp. Soc., 2003.
12. D. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2), 1993.
13. J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proc. 8th ACM Conference on Computer and Communications Security (CCS'01)*, 2001.
14. J. Millen and V. Shmatikov. Symbolic protocol analysis with products and Diffie-Hellman exponentiation. In *Proc. 16th Computer Security Foundation Workshop (CSFW'03)*, pages 47–62. IEEE Comp. Soc. Press, 2003.
15. M. Rusinowitch and M. Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theoretical Computer Science*, 1-3(299):451–475, 2003.

A Proofs of Section 2

A.1 More Examples

Lemma 14. *The inference system described in Example 4 is good.*

Proof. In addition to the bad proofs that we have seen in Example 3, we have now the following types of bad proofs:

$$\dfrac{\dfrac{u \quad v}{\mathtt{sign}(u,v)}\,(\mathsf{S})}{u}\,(\mathsf{C}) \qquad \dfrac{\dfrac{b_n(u,v_1 \dots, v_n) \quad s}{\mathtt{sign}(b_n(u,v_1,\dots,v_n),s)}\,(\mathsf{S}) \qquad v_1 \ \dots \ v_n}{\mathtt{sign}(u,s)}\,(\mathsf{UB}_2)$$

$$\dfrac{\dfrac{u \quad v}{\mathtt{blind}(u,v)}\,(\mathsf{B}) \qquad v}{u}\,(\mathsf{UB}_1) \qquad \dfrac{\dfrac{\mathtt{sign}(b_n(u,v_1,\dots,v_n),s) \quad v_1 \ \dots \ v_n}{\mathtt{sign}(u,s)}\,(\mathsf{UB}_2)}{u}\,(\mathsf{C})$$

$$\dfrac{\dfrac{\mathtt{sign}(b_{n+m}(v,t_1,\dots,t_n,v_1,\dots,v_m),s) \quad v_1 \ \dots \ v_m}{\mathtt{sign}(b_n(v,t_1,\dots,t_n),s)}\,(\mathsf{UB}_2) \qquad t_1 \ \dots \ t_n}{\mathtt{sign}(v,s)}\,(\mathsf{UB}_2)$$

In the first two cases ($\mathsf{S} - \mathsf{C}$ and $\mathsf{B} - \mathsf{UB}_1$), we have as in Example 3 corresponding trivial proofs, which are smaller w.r.t. any well-founded extension of subterm ordering.

Assume now that the bad proof π is as in case $\mathsf{S} - \mathsf{UB}_2$. Then let π' be the proof consisting of several applications of UB_1 followed by an application of S. We obviously have $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. Moreover, we have that

$$\begin{aligned}
&\mu(\pi') = \{\{b_n(u,v_1,\dots,v_n)\},\ \{b_{n-1}(u,v_1,\dots,v_{n-1})\},\ \dots,\ \{b_1(u,v_1)\},\ \{\mathtt{sign}(u,s)\}\}\\
&\mu(\pi) = \{\{\mathtt{sign}(b_n(u,v_1,\dots,v_n),s)\},\{\mathtt{sign}(b_n(u,v_1,\dots,v_n),s),\mathtt{sign}(u,s)\}\}
\end{aligned}$$

and hence $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$ for any total well-founded extension $\prec$ of the subterm ordering.

When the bad proof is as in the case $\mathsf{UB}_2 - \mathsf{C}$, the situation is rather similar.

Lastly, assume that the bad proof π is as in case $\mathsf{UB}_2 - \mathsf{UB}_2$. Then let π' be

$$\dfrac{\mathtt{sign}(b_{n+m}(v,t_1,\dots,t_n,v_1,\dots,v_m),s) \ \ t_1 \ \dots \ t_n \ \ v_1 \ \dots \ v_m}{\mathtt{sign}(v,s)}\,(\mathsf{UB}_2)$$

Again, we obviously have $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. Moreover, we have

$$\begin{aligned}
&\mu(\pi') = \{\{\mathtt{sign}(b_{n+m}(v,t_1,\dots,t_n,v_1,\dots,v_m),s),\mathtt{sign}(v,s)\}\}\\
&\mu(\pi) = \left\{ \begin{array}{l} \{\mathtt{sign}(b_{n+m}(v,t_1,\dots,t_n,v_1,\dots,v_m),s),\mathtt{sign}(b_n(v,t_1,\dots,t_n),s)\}, \\ \quad \{\mathtt{sign}(b_n(v,t_1,\dots,t_n),s),\mathtt{sign}(v,s)\} \end{array} \right\}
\end{aligned}$$

Hence, to ensure $\mu(\pi') \ (\prec_m)_m \ \mu(\pi)$, it is sufficient to consider a total well-founded extension $\prec$ of the subterm ordering such that for all $n, v, s, t_1, \ldots, t_n$, we have that $\mathtt{sign}(v, s) \prec \mathtt{sign}(b_n(v, t_1, \ldots, t_n), s)$. $\square$

Example 9. We consider another property satisfied by encryption, using block ciphers as Electronic Code Book (ECB). ECB consists in partitioning the message into blocks and encrypting each of them separately. This homomorphism property can be formalized using the following rules:

$$\frac{\mathtt{enc}(x, \langle y, z\rangle, r)}{\mathtt{enc}(x, y, r)} \qquad \frac{\mathtt{enc}(x, \langle y, z\rangle, r)}{\mathtt{enc}(x, z, r)}$$

This inference system is not good. However, if we let $P(x)$ be the least set of terms such that $x \in P(x)$ and $t \in P(x) \Rightarrow \langle t, u\rangle \in P(x) \wedge \langle u, t\rangle \in P(x)$, then the following infinite set of rules, together with the rules of Example 1, is good.

$$\frac{\mathtt{enc}(x, t, r)}{\mathtt{enc}(x, y, r)}\,(\mathsf{R_t}) \quad \text{for any } t \in P(y)$$

Lemma 15. *The inference system described in Example* **??** *is good.*

Proof. Again, we notice that, w.r.t. to Example 3, there are only three additional types of bad proofs:

$$\dfrac{\dfrac{\mathtt{enc}(t, u, r)}{\mathtt{enc}(t, v, r)}\,(\mathsf{R_t})}{\mathtt{enc}(t, w, r)}\,(\mathsf{R_t}) \qquad \dfrac{\dfrac{t \quad u \quad r}{\mathtt{enc}(t, u, r)}\,(\mathsf{E})}{\mathtt{enc}(t, v, r)}\,(\mathsf{R_t}) \qquad \dfrac{\dfrac{\mathtt{enc}(\mathtt{pub}(t), u, r)}{\mathtt{enc}(\mathtt{pub}(t), v, r)}\,(\mathsf{R_t}) \qquad \mathtt{priv}(t)}{v}\,(\mathsf{D})$$

where $u \in P(v)$ and $v \in P(w)$.

Let π be a bad proof of the first type. Then, by definition of P, we have $u \in P(w)$ and hence the following proof π' of $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$

$$\frac{\mathtt{enc}(t, u, r)}{\mathtt{enc}(t, w, r)}\,(\mathsf{R_t})$$

Obviously, $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. We have $\mu(\pi') = \{\{\mathtt{enc}(t, u, r), \mathtt{enc}(t, w, r)\}\}$ and $\mu(\pi) = \{\{\mathtt{enc}(t, u, r), \mathtt{enc}(t, v, r)\}, \{\mathtt{enc}(t, v, r), \mathtt{enc}(t, w, r)\}\}$. To have that $\mu(\pi') \ (\prec_m)_m \ \mu(\pi)$, we must choose an extension $\prec$ of subterm ordering s.t. $\mathtt{enc}(t, w, r) \prec \mathtt{enc}(t, v, r)$, for all t, r, w, v with $v \in P(w)$.

Let π be a bad proof of the second type. Since $u \in P(v)$, we have $u \vdash v$, by using only Proj_1 and Proj_2. Let π_{uv} be this proof of $u \vdash v$ and let π' be the following proof of $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$

$$\frac{t \quad \pi_{uv} \quad r}{\mathtt{enc}(t, v, r)}\,(\mathsf{E})$$

Obviously, we have that $\texttt{leaves}(\pi') \subseteq \texttt{leaves}(\pi)$. We have also that $\mu(\pi') = \{\{u\}, \{u_1\}, \dots, \{u_n\}, \{\texttt{enc}(t, v, r)\}\}$, where for all $1 \leq i \leq n$, $u_i \in st(u)$. On the other hand, $\mu(\pi) = \{\{\texttt{enc}(t, u, r)\}, \{\texttt{enc}(t, u, r), \texttt{enc}(t, v, r)\}\}$. Hence, for any extension $\prec$ of the subterm ordering we have $\mu(\pi') \ (\prec_m)_m \ \mu(\pi)$.

Lastly, let π be a bad proof of the third type. Since $u \in P(v)$. we have $u \vdash v$ by using only Proj_1 and Proj_2. Let π_{uv} be such a proof. Consider then the proof π' of $\texttt{leaves}(\pi) \vdash \texttt{conc}(\pi)$ which consists of applying (D) on the premises $\texttt{enc}(\texttt{pub}(t), u, r)$, and $\texttt{priv}(t)$, followed by the proof π_{uv}.

We have $\texttt{leaves}(\pi') \subseteq \texttt{leaves}(\pi)$, and

- $\mu(\pi') = \{\{\texttt{enc}(\texttt{pub}(t), u, r), \texttt{priv}(t)\}, \{u\}, \{u_1\}, \dots \{u_n\}\}$,
- $\mu(\pi) = \{\{\texttt{enc}(\texttt{pub}(t), u, r), \texttt{enc}(\texttt{pub}(t), v, r)\}, \{\texttt{enc}(\texttt{pub}(t), v, r), \texttt{priv}(t)\}\}$.

Since $u_1, \dots, u_n \in st(u)$, to have $\mu(\pi') \ (\prec_m)_m \ \mu(\pi)$ it is sufficient to choose an extension $\prec$ of subterm ordering such that $\texttt{priv}(t) \prec \texttt{pub}(t)$, for all t. Note that this restriction of the ordering is not in contradiction with the one that was required for avoiding the first type of bad proofs.

We conclude that there is a total and well-founded extension $\prec$ of subterm ordering that makes the system ECB good. $\square$

A.2 Subterm convergent theories

The goal of this section is to show that any intruder theory that comes from a finite subterm-convergent equational theory satisfy our hypotheses.

We assume a finite term rewriting system $\mathcal{R} = \{l_1 \to r_1, \dots, l_n \to r_n\}$ such that every r_i is a strict subterm of l_i and such that the associated reduction relation $\to_{\mathcal{R}}$ is confluent. In that case, every term t has a unique normal form $t\downarrow$ w.r.t. $\mathcal{R}$. The set of function symbols is split into $\mathcal{F}_{\texttt{pub}}$ and $\mathcal{F}_{\texttt{priv}}$. We let $\mathcal{I}$ be the following inference system:

$$\mathcal{I} = \begin{cases} \dfrac{l_1 \quad \cdots \quad l_n}{r} & \text{if } f(l_1, \dots, l_n) \to r \in \mathcal{R} \text{ for some } \mathsf{f} \in \mathcal{F}_{\texttt{pub}}, \\ & \qquad \text{and } l_1, \dots, l_n \text{ are in normal form} \\ \dfrac{x_1 \quad \cdots \quad x_n}{\mathsf{f}(x_1, \dots, x_n)} & \text{if } f \in \mathcal{F}_{\texttt{pub}} \end{cases}$$

Lemma 16. *Let $t_1, \dots, t_n, t$ be terms in normal form. There is a context $C \in T(\mathcal{F}_{\texttt{pub}}, \mathcal{X})$ such that $C\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}\downarrow = t$ if, and only if, $t_1, \dots, t_n \vdash t$ in $\mathcal{I}$.*

Proof. We prove that for any context C such that $C\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}\downarrow = t$, there is a proof $t_1, \dots, t_n \vdash t$ in $\mathcal{I}$. We use an induction on C. If C is empty, then this holds trivially. If $C = \mathsf{f}(C_1, \dots, C_m)$ with $\mathsf{f} \in \mathcal{F}_{\texttt{pub}}$, we apply the induction hypothesis. We have that $t_1, \dots, t_n \vdash u_1, \ \dots, \ t_1, \dots, t_n \vdash u_m$ in $\mathcal{I}$ where, for every i, $u_i = C_i[t_1, \dots, t_m]\downarrow$.

- Either $t = \mathsf{f}(u_1, \ldots, u_m)$ and we can simply apply the rule $I(x_1), \ldots, I(x_m) \to I(\mathsf{f}(x_1, \ldots, x_m)$;
- or else there is a redex: $f(u_1, \ldots, u_m) = l\sigma$ for some rule $l \to r \in \mathcal{R}$. Let then $l = \mathsf{f}(l_1, \ldots, l_m)$: there is a substitution σ such that $l_i\sigma = u_i$ for every i. Furthermore, since u_i is in normal form, each l_i is in normal form, as well as r that is a subterm of some l_i. Hence we can apply a rule $I(l_1), \ldots, I(l_m) \to I(r)$ and get the term $r\sigma$, that is in normal form. Then $t_1, \ldots, t_n \vdash r\sigma = t$ in $\mathcal{I}$.

Conversely, if $t_1, \ldots, t_n \vdash t$ in $\mathcal{I}$, we build the context C by induction on the proof size. If the proof is trivial, i.e. $t_i = t$ for some i, we can choose $C = x_i$. Otherwise, we distinguish two cases:

- If the last proof step is $\dfrac{u_1 \ \cdots \ u_m}{\mathsf{f}(u_1, \ldots, u_m)}$ by induction hypothesis there are contexts $C_1, \ldots, C_m$ such that $C_i[t_1, \ldots, t_m]\!\downarrow = u_i$ for every i and it suffices to choose $C = \mathsf{f}(C_1, \ldots, C_m)$.
- If the last proof step is $\dfrac{l_1\sigma \ \cdots \ l_m\sigma}{r\sigma}$ and, for some function symbol $\mathsf{f} \in \mathcal{F}_{\mathtt{pub}}$, $\mathsf{f}(l_1, \ldots, l_m) \to r \in \mathcal{R}$ and $l_1, \ldots, l_m$ are in normal form, then, by induction hypothesis, we let C_i, $i = 1, ..., m$ be such that $C_i[t_1, \ldots, t_n]\!\downarrow = l_i\sigma$ and $C = \mathsf{f}(C_1, \ldots, C_m)$. Then, we have that:

$$C[t_1, \ldots, t_n]\!\downarrow = \mathsf{f}(l_1\sigma, \ldots, l_n\sigma)\!\downarrow = r\sigma\!\downarrow = r\sigma = t.$$

This allows us to conclude. □

Now, we repeatedly perform the following transformations on the set of inference rules $\mathcal{I}$:

1. we remove from $\mathcal{I}$ the inference rules whose conclusion also belongs to the premises
2. we replace inference rules $I(s_1) \ldots, I(x), \ldots, I(s_n) \to I(t)$ with the inference rules $I(s_1), \ldots, \ldots, I(s_n) \to I(t)$ (i.e. remove $I(x)$ from the premises) when x is a variable that neither occurs in t nor as a strict subterm of some s_i.
3. for every public symbol g and every rule $\dfrac{l_1 \ \cdots \ l_k \ \ \mathsf{g}(u_1, \ldots, u_m) \ \ \cdots l_n}{r}$ we add the inference rule $\dfrac{l_1 \ \cdots \ l_k \ \ u_1 \ \cdots \ u_m \ \ \cdots l_n}{r}$

until getting a fixed point.

Since the size (number of function symbols occurring in it) of new inference rules is strictly smaller than the size of some inference rule in the original set, the fixed point is reached in a finite number of steps.

Lemma 17. *The transformation steps preserve the deducibility relation.*

Proof. Indeed, every new inference rule can be proved using the other inference rules and every inference rule that is removed (or replaced) can also be proved using the other rules. □

As a result, all inference rules are either compositions $I(x_1), \dots, I(x_n) \to I(\mathsf{f}(x_1, \dots, x_n))$ or decomposition rules $I(u_1), \dots, I(u_n) \to I(r)$: in the above procedure, we always keep as an invariant that the conclusion of an inference rule, which is not a composition, is a subterm of some of the premises. There is no versatile rule and our additional hypotheses are satisfied. It remains to show that the resulting inference system is good.

Lemma 18. *The resulting inference system is good.*

Proof. Since maximal premises of a decomposition rule cannot be variables (2[nd] transformation in the above computation), the only possible bad patterns are of the form:

$$\frac{u_1 \quad \cdots \quad u_k \quad \dfrac{x_1\sigma \quad \cdots x_m\sigma}{\mathsf{g}(x_1, \dots, x_m)\sigma} \quad u_{k+2} \quad \cdots \quad u_n}{r}$$

where $\mathsf{g} \in \mathcal{F}_{\mathtt{pub}}$, $u_{k+1} = \mathsf{g}(x_1\sigma, \dots, x_m\sigma)$. Thanks to the above completion (3[rd] step), there is then a simpler proof using the inference rule

$$\frac{u_1 \quad \cdots \quad u_k \quad x_1\sigma \quad \cdots \quad x_m\sigma \quad u_{k+2} \quad \cdots \quad u_n}{r}$$

This allows us to conclude. □

A.3 Some properties of good inference systems

Lemma 1. *Consider a good inference system. Let $H_1 \subseteq H_2 \subseteq \cdots \subseteq H_n$ be an increasing sequence of sets of terms and $i \in \{1, \dots, n\}$. If π is a proof of $H_i \vdash u$ then there is a simple proof of $H_i \vdash u$.*

Proof. Let $\prec$ be the total well-founded extension of $\lhd$ witnessing the fact that the inference system is good. Let π be a proof of $H_i \vdash u$ and $H_0 = \emptyset$. We will use induction on the pair $(\mathsf{M}(\pi), \mu(\pi))$ (considering the lexicographic ordering) where $\mathsf{M}(\pi)$ is the multiset $\{i \mid u \in \mathtt{leaves}(\pi) \text{ and } u \in H_i \smallsetminus H_{i-1}\}$, to prove that there is a simple proof π' of $H_i \vdash u$ with $(\mathsf{M}(\pi'), \mu(\pi')) \leq_{lex} (\mathsf{M}(\pi), \mu(\pi))$.

Base case: $\mathsf{M}(\pi) = \{1, \dots, 1\}$. If π does not contain any bad pattern, then π is simple and we easily conclude. Otherwise, let π_1 be a bad pattern of π. We have that:

$$\pi_1 \;=\; \left\{ \frac{\pi_2^1 \quad \cdots \quad \pi_2^{i-1} \qquad \dfrac{\pi_1^1 \quad \cdots \quad \pi_1^m}{\pi_2^i}\,\mathsf{R}_1 \qquad \pi_2^{i+1} \quad \cdots \quad \pi_2^n}{v}\,\mathsf{R}_2 \right.$$

such that $\widehat{\pi_1}$ defined below is a bad proof.

$$\frac{\mathtt{conc}(\pi_2^1) \cdots \mathtt{conc}(\pi_2^{i-1}) \qquad \dfrac{\mathtt{conc}(\pi_1^1) \cdots \mathtt{conc}(\pi_1^m)}{\mathtt{conc}(\pi_2^i)}\,\mathsf{R}_1 \qquad \mathtt{conc}(\pi_2^{i+1}) \cdots \mathtt{conc}(\pi_2^n)}{v}\,\mathsf{R}_2$$

By hypothesis, we know that there is a proof $\widehat{\pi_2}$ such that $\texttt{conc}(\widehat{\pi_2}) = v$ and $\texttt{leaves}(\widehat{\pi_2}) \subseteq \texttt{leaves}(\widehat{\pi_1})$ (multiset inclusion) and $\mu(\widehat{\pi_2})(\prec_m)_m\mu(\widehat{\pi_1})$. Let π_2 be the proof $\widehat{\pi_2}$ in which each hypothesis $\texttt{conc}(\pi_i^j)$ is replaced with the corresponding proof π_i^j.

Let $M = \biguplus_{1 \leq j \leq m} \mu(\pi_1^j) \uplus \biguplus_{1 \leq j < i} \mu(\pi_2^j) \uplus \biguplus_{i < j \leq n} \mu(\pi_2^j)$. We have that:

$$\mu(\pi_2)\ (\prec_m)_m\ M \uplus \mu(\widehat{\pi_2})\ (\prec_m)_m\ M \uplus \mu(\widehat{\pi_1}) = \mu(\pi_1).$$

Note that the first inequality is a consequence of $\texttt{leaves}(\widehat{\pi_2}) \subseteq \texttt{leaves}(\widehat{\pi_1})$ (multiset inclusion). Then, let π' be the proof obtained by replacing π_1 with π_2 in π. We have that $\mathsf{M}(\pi') \subseteq \mathsf{M}(\pi)$ (multiset inclusion) and $\mu(\pi') \prec \mu(\pi)$, thus we may apply our induction hypothesis in order to conclude.

Induction step: $\mathsf{M}(\pi)$ *contains at least an element* $i > 1$. First, if there exists a proof π' of u such that $\mathsf{M}(\pi') <_m \mathsf{M}(\pi)$ then we easily conclude by relying on our induction hypothesis. Otherwise either π is reduced to a leaf and it is easy to see that π is left-minimal (since it does not exist a proof π' of u such that $\mathsf{M}(\pi') <_m \mathsf{M}(\pi)$), or we have that π is of the form:

$$\pi = \left\{ \quad \frac{\pi_1 \quad \cdots \quad \pi_n}{v}\ \mathsf{R} \right.$$

We can apply our induction hypothesis on each π_i. For each i, we deduce that there exists a simple proof π_i' such that $\texttt{conc}(\pi_i') = \texttt{conc}(\pi_i)$ and we have also that $(\mathsf{M}(\pi_i'), \mu(\pi_i')) \leq_{lex} (\mathsf{M}(\pi_i), \mu(\pi_i))$. Let π' be the proof obtained by applying the inference rule R on $\pi_1', \ldots, \pi_n'$. Note that $(\mathsf{M}(\pi'), \mu(\pi')) \leq_{lex} (\mathsf{M}(\pi), \mu(\pi))$.

If the resulting proof π' is not a simple proof of $H_i \vdash u$, this means that π' ends with a bad pattern. In such a case, we can replace it as in the previous case. We obtain a proof π'' of u such that $(\mathsf{M}(\pi''), \mu(\pi'')) \leq_{lex} (\mathsf{M}(\pi'), \mu(\pi'))$. This allows us to conclude. □

Lemma 2 (locality). *Let π be a proof of $H \vdash u$ without bad pattern, one of the following occurs:*

- *$\texttt{last}(\pi)$ is a composition and $\texttt{step}(\pi) \subseteq st(H \cup \{u\})$;*
- *π is reduced to a leaf or $\texttt{last}(\pi)$ is a decomposition and $\texttt{step}(\pi) \subseteq st(H)$;*
- *$\texttt{last}(\pi)$ is (an instance of) a versatile rule and $\texttt{step}(\pi') \subseteq st(H)$ for any strict subproof π' of π.*

Proof. Let π be a proof of $H \vdash u$ without bad pattern. We prove the result by induction on π.

Base case: π is reduced to a leaf. The result obviously holds.

Induction step. If π ends with an instance of a composition rule, then we easily conclude. Otherwise, π ends with an instance of a versatile or a decomposition rule. We have that

$$\pi = \left\{ \quad \frac{\pi_1 \quad \cdots \quad \pi_n}{u}\ \mathsf{R} \right. \text{ with } \mathsf{R} \;=\; \frac{v_1 \quad \cdots \quad v_n}{v_0}$$

Let $\mathtt{conc}(\pi_i) = u_i$ and $I = \{i \in \{0, 1, \ldots, n\} \mid v_i \in \mathtt{Max}(\mathsf{R})\}$. For every $i \in \{1, \ldots, n\}$, let R_i be the last inference rule used in π_i. By definition of a proof without bad pattern, we have that π_i does not end with an instance of a composition or a versatile rule for any $i \in I \smallsetminus \{0\}$. Thus, we can apply our induction hypothesis, we have that $\mathtt{step}(\pi_i) \subseteq st(H)$ for every $i \in I \smallsetminus \{0\}$. Now, let $J = \{1, \ldots, n\} \smallsetminus I$ and $j \in J$, we have that $v_j \in st(v_i)$ for some $i \in I$. Actually, we have that $i \neq 0$. This holds by definition of a decomposition rule when R is a decomposition and thanks to our additional hypothesis (Condition 2b) when R is versatile. Hence for any $j \in J$, we have that $u_j \in st(u_i)$ for some $i \in I \smallsetminus \{0\}$. Moreover, by induction hypothesis, $\mathtt{step}(\pi_j) \subseteq st(H \cup \{u_j\})$. Hence we have that $\mathtt{step}(\pi_i) \subseteq st(H)$ for any $i \in \{1, \ldots, n\}$. This allows us to conclude. □

The following example shows that Condition 2b is crucial to establish Lemma 2.

Example 10. Consider the inference system made up of the two rules described below. The first one is a composition rule whereas the second one is a versatile rule where $\mathtt{Max}(\mathsf{R}_2) = \{\mathsf{h}(x), \mathsf{f}(\mathsf{g}(x))\}$.

$$\frac{x}{\mathsf{g}(x)}\ \mathsf{R}_1 \qquad\qquad \frac{\mathsf{h}(x) \quad \mathsf{g}(x)}{\mathsf{f}(\mathsf{g}(x))}\ \mathsf{R}_2$$

Let $H = \{a, \mathsf{h}(a)\}$. We have that $H \vdash \mathsf{f}(\mathsf{g}(a))$. The proof witnessing this fact ends with an instance of the versatile rule R_2. However, the premise $\mathsf{g}(a) \notin st(H)$. We do not consider such an inference system since the premise $\mathsf{g}(x)$ is not maximal in R_2, thus $\mathsf{g}(x)$ has to be a strict subterm of another premise. This is not the case.

B Proofs of Section 4

B.1 Soundness

Lemma 3 (soundness). *Let D be a constraint system such that $D \rightsquigarrow D'$, then D' is a constraint system and $Sol(D') \subseteq Sol(D)$.*

Proof. Let $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ be a constraint system and D' be such that $D \rightsquigarrow D'$. We have to show that D' is a constraint system. First $D' = \exists \widetilde{z'}.[C' \mid \mathsf{E}']$ where:

- $\widetilde{z'}$ is a sequence of variables;
- E' is a set of equations in solved form;
- C' is a conjunction of deducibility constraints and since the substitution involved is applied to the constraints in C, we have that $var(C') \cap dom(\mathsf{E}') = \emptyset$. It is also easy to show that monotony is preserved. Concerning origination, we consider each rule in turn.

Notice first that origination is preserved by application of a substitution, i.e. if C satisfies origination, then $C\theta$ satisfies origination for any substitution θ.

Rule Axiom. In such a case we have that $C' = C\theta \smallsetminus \{H\theta \Vdash u\theta\}$, for some $H \Vdash u \in C$ such that $u\theta \in H\theta$. Since $C\theta$ satisfies origination and no variable can be introduced in $u\theta$, we conclude that C' satisfies origination.

Rule Triv. In such a case we have that $C' = C \smallsetminus \{H' \Vdash x\}$, where $H \Vdash x \in C$ for some $H \subseteq H'$. The constraint that we removed does not introduce any new variable. This allows us to easily conclude.

Rule Comp. In such a case we have that:

$$C' = C \smallsetminus \{H \Vdash \mathsf{f}(t_1, \ldots, t_n)\} \cup \{H \Vdash t_1, \ldots, H \Vdash t_n\}.$$

It is easy to see that origination is satisfied by C'.

Rule Dec. In such a case have that

$$C' = (C\theta \smallsetminus \{H\theta \Vdash v\theta\}) \cup \{H\theta \Vdash u_i\theta \mid 1 \leq i \leq n\} \cup \{H'\theta \Vdash v_j\theta \mid 1 \leq j \leq m\}$$

where $\theta = \mathsf{mgu}(\langle w, w_1, \ldots, w_n\rangle, \langle v, u_1, \ldots, u_n\rangle)$, $u_1, \ldots, u_n \in st(H) \smallsetminus \mathcal{X}$. Moreover, the variables that occur in the terms $w, w_1, \ldots, w_n$ are fresh and we have that $var(w) \subseteq var(w_1, \ldots, w_n)$.

By our first observation, $C\theta$ satisfies the origination property. The new constraints introduced do not cause any trouble. However, to safely remove $H\theta \Vdash v\theta$, we have to check that $v\theta = w\theta$ does not introduce any new variable for the first time. Let $x \in var(v\theta)$. We have that $var(v\theta) \subseteq var(\{v_1\theta, \ldots, v_m\theta, w_1\theta, \ldots, w_n\theta\})$. If $x \in var(v_i\theta)$, then we easily conclude. Otherwise, we have that $x \in var(w_i\theta) = var(u_i\theta)$ and we easily deduce that $x \in var(H\theta)$ since $u_i \in st(H)$. This allows us to conclude that C' satisfies the origination property.

By considering each transformation rule in turn is is easily to show that if $D \rightsquigarrow D'$ and $\sigma \in Sol(D')$, then $\sigma \in Sol(D)$. This allows us to conclude the proof. $\square$

Lemma 4. *Let D be a constraint system that is uniquely determined and D' be such that $D \rightsquigarrow D'$. Then D' is uniquely determined.*

Proof. Let $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ and $D' = \exists \tilde{z'}.[C' \mid \mathsf{E}']$ be two constraint systems such that $D \rightsquigarrow D'$. We consider each transformation rule in turn.

- Rule Triv or Comp. In such a case, we have that $\tilde{z'} = \tilde{z}$ and $\mathsf{E}' = \mathsf{E}$ and we easily conclude.
- Rule Axiom. Then $\tilde{z'} = \tilde{z}$ and $\mathsf{E}' = \mathsf{E} \cup \theta$, where θ is a set of equations such that $var(\theta) \subseteq var(C)$. Let ρ be any ground substitution with $dom(\rho) = fvar(D') = fvar(D)$ and assume that $\mathsf{mgu}(\mathsf{E}'\rho) \neq \perp$. Then, since $\mathsf{E}' = \mathsf{E} \cup \theta$, we have that $\mathsf{mgu}(\mathsf{E}\rho) \neq \perp$. By using the fact that D is uniquely determined, we deduce that $\tau = \mathsf{mgu}(\mathsf{E}\rho)$ is a ground substitution. Hence $\mathsf{mgu}(\mathsf{E}'\rho) = \mathsf{mgu}(\mathsf{E}\rho) = \tau$. This allows us to conclude.

- Rule Dec. In such a case, we have that $\tilde{z'} = \tilde{z} \cup \tilde{x}$ and $\mathsf{E}' = \mathsf{E} \cup \theta$, where $\theta = \mathsf{mgu}(u, v)$ for some term u, v such that $var(u) \subseteq var(C)$ and $var(v) = \tilde{x}$. Let ρ be any ground substitution with $dom(\rho) = fvar(D') = fvar(D)$ and assume that $\mathsf{mgu}(\mathsf{E}'\rho) \neq \bot$. As in the previous case, we deduce that $\tau = \mathsf{mgu}(\mathsf{E}\rho)$ is a ground substitution. On the other hand, we have that

$$\mathsf{mgu}(\mathsf{E}'\rho) = \mathsf{mgu}(\mathsf{E}\rho \cup \{u\rho = v\}) = \tau \cup \mathsf{mgu}(u(\rho \cup \tau) = v).$$

  It is easy to see that this substitution is a ground substitution whose domain is $\tilde{z} \cup \tilde{x}$. This is due to the fact that $\tilde{x} = var(v)$ and $u(\rho \cup \tau)$ is a ground term. □

B.2 Solved forms

Lemma 5. *A solved form has always at least one solution.*

Proof. Let u_0 be any ground term in H_1 (such a term exists since $H_1 \neq \emptyset$) and θ_{E} be the substitution associated to the set of equations E. Let δ be the substitution such that

$$dom(\delta) = \{x_1, \ldots, x_n\} \cup var(img(\theta_{\mathsf{E}})) \text{ and } x\delta = u_0 \text{ for each } x \in dom(\delta).$$

Let σ be a substitution such that $dom(\sigma) = var(D)$ and

- $x\sigma = (x\theta_{\mathsf{E}})\delta$ for any $x \in dom(\theta_{\mathsf{E}})$;
- $x\sigma = x\delta$ otherwise.

Note that $\sigma \in Sol([H_1 \Vdash x_1 \wedge \ldots \wedge H_n \Vdash x_n \mid \mathsf{E}])$. Now, let $\sigma' = \sigma|_{fvar(D)}$. It is easy to see that $\sigma' \in Sol(D)$. □

B.3 Completeness

Lemma 6. *Let D be a constraint system of the form $[C \mid \mathsf{E}]$. Let $H \in \mathsf{LH}(D)$ be such that for every $y \in var(H)$ there is a constraint $H_y \Vdash y \in D$. Let σ be a solution of D and v be a term such that $H\sigma \vdash v$. Let $u \in st(v)$. Then:*

1. *either $u \in (st(H) \smallsetminus \mathcal{X})\sigma$;*
2. *or $H\sigma \vdash u$ and any simple proof π of $H\sigma \vdash u$ ends with a composition or a versatile rule.*

Proof. Let Π be the set of simple proofs of $H\sigma \vdash v$. We prove this result by induction on the pair (H, d) where d is the size of a minimal proof in Π. We distinguish two cases depending on the last inference rules of the proofs in Π.

- *There is a $\pi \in \Pi$ that ends with a decomposition or an axiom.* In such a case, thanks to Lemma 2 (and trivially, in case of an axiom) we have that $v \in st(H\sigma)$, and thus $u \in st(H\sigma)$. Either $u \in (st(H) \smallsetminus \mathcal{X})\sigma$ and we easily conclude (this always happens when H is ground, our base case). Otherwise, $u \in st(y\sigma)$ for some $y \in var(H)$. In such a case, we have that $u \in st(y\sigma)$ and $H_y\sigma \vdash y\sigma$ with $H_y \subsetneq H$. Hence, we can apply our induction hypothesis and we easily conclude.
- *Each proof $\pi \in \Pi$ ends with a composition or a versatile rule.* Let us choose $\pi \in \Pi$ to be minimal in size among the proofs in Π. Let $\pi_1, \dots, \pi_n$ be the immediate subproofs of π. We assume that the roots of $\pi_1, \dots, \pi_n$ are labelled with $u_1, \dots, u_n$ respectively. If $u = v$ then we easily conclude. Otherwise, *since we assumed that the strict subterms of the conclusion are subterm of the premises* (Condition 2a), we have that $u \in st(u_j)$ for some j such that $1 \leq j \leq n$. We have that $H\sigma \vdash u_j$ and $u \in st(u_j)$. Hence, we can apply our induction hypothesis since π_j is a simple proof of $H\sigma \vdash u_j$ and the size of π_j is smaller than the size of π. Then, we easily conclude. $\square$

Lemma 7. *If D is a constraint system that is uniquely determined and $\sigma \in Sol(D)$, then either D is in solved form or else there is a D' such that $D \rightsquigarrow D'$, $\sigma \in Sol(D')$, and $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$.*

Proof. Let $D = \exists \tilde{z}.[C \mid \mathsf{E}]$. Assume first that the constraint system D is not in solved form and every deducibility constraint is of the form $H \Vdash x$, where x is a variable, then we can apply Triv, yielding a system D' such that $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$. Now, assume that there is a deducibility constraint whose right member is not a variable. Consider one, say $H \Vdash s$, such that s is not a variable, and whose left member H is minimal w.r.t. inclusion. Since σ is a solution of D, we know that there exists a ground substitution τ with $dom(\tau) = \tilde{z}$ and such that $\sigma \cup \tau$ satisfies each elementary constraint in C and E. Actually, since D is uniquely determined, we know that such a substitution τ is uniquely defined. Let $\overline{\sigma} = \sigma \cup \tau$. Let π be a simple proof of $H\overline{\sigma} \vdash s\overline{\sigma}$ that is minimal in size (note that such a proof exists thanks to Lemma 1). We distinguish three cases, depending on the last rule of π.

- If π is reduced to a leaf, then we can apply Axiom, yielding a constraint system D' such that $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$ since we removed a deducibility constraint.
- If π ends with an instance of a composition rule, then we may apply Comp, yielding a constraint system D'. We have that σ is a solution of D' and $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$.
- If π ends with an instance of a decomposition or a versatile rule, say

$$\mathsf{R} = \frac{v_1 \quad \dots \quad v_m}{w}$$

  then we show that we may apply Dec. Let θ be the ground substitution such that $dom(\theta) = var(\mathsf{R})$ and $\mathsf{R}\theta$ corresponds to the instance used in the proof π. We have that $w\theta = s\overline{\sigma}$ and $v_1\theta, \ldots, v_m\theta$ are the instances of the premises corresponding to the proof. Let $\pi_1, \ldots, \pi_m$ be the immediate subproofs of π. Since π is simple and thus without bad pattern, by Lemma 2 we have that $v_i\theta \in st(H\overline{\sigma})$ for every i. Let $1 \leq i \leq n$. We have that:

1. either $v_i\theta \in (st(H) \smallsetminus \mathcal{X})\overline{\sigma}$,
2. or $v_i\theta \in st(x\overline{\sigma})$ for some $x \in var(H)$. By the choice of H and origination, we have that $H_x\overline{\sigma} \vdash x\overline{\sigma}$, with $H_x \subsetneq H$. Thanks to Lemma 6 (applied to $[C \mid \mathsf{E}]$, $H_x\overline{\sigma}$, $x\overline{\sigma}$ and $v_i\theta$), we deduce that π_i is a simple proof of $H_x\overline{\sigma} \vdash v_i\theta$ and π_i ends with an instance of a versatile or a composition rule. Hence, those premises that are not in $(st(H) \smallsetminus \mathcal{X})\overline{\sigma}$ can be proved with a strict smaller left hand side $H_x \subsetneq H$.

Moreover, if v_i is maximal among $\{v_1, \ldots, v_m, w\}$ then $v_i\theta \in (st(H) \smallsetminus \mathcal{X})\overline{\sigma}$. Indeed, in such a case, since π does not contain any bad pattern, π_i can not end with an instance of a versatile or a composition rule. Thus, the second case is impossible. Then, it follows that $D \rightsquigarrow D'$ with the Dec rule and σ is a solution of D'. Furthermore $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$. This allows us to conclude. □

C Termination

Lemma 9 (soundness). *Let D be an extended constraint system such that $D \mapsto D'$ then D' is an extended constraint system and $Sol(D') \subseteq Sol(D)$.*

Proof. By relying on the previous set of transformation rules, it is easy to see that $(\mathsf{Act}(D') \cup \mathsf{Fr}(D'))\theta'$ where $\theta' = \mathsf{mgu}(\mathsf{E}(D'))$ is a constraint system. The main point we have to show is the fact that $\mathsf{Act}(D')$ is a constraint system. We distinguish three cases:

Rule Active. The result is immediate since then $\mathsf{Act}(D) \rightsquigarrow \exists\tilde{x}.[\mathsf{Act}(D') \mid \theta]$, for some variables $\tilde{x}$, and some substitution θ. Lemma 3 allows us to conclude.

Rule Freeze. In such a case, $\mathsf{Act}(D') = \mathsf{Act}(D) \smallsetminus \{H \Vdash v\} \cup \{H \Vdash u_1, \ldots, H \Vdash u_n\}$ and we have that $H_x \subsetneq H$ for every $x \in var(v)$. This last fact ensures that origination holds also for $\mathsf{Act}(D')$.

Rule Open. In such a case, we have that $\mathsf{Act}(D') = (\mathsf{Act}(D) \cup \mathsf{Fr}(D))\theta$ where $\theta = \mathsf{mgu}(\mathsf{E})$. This is a constraint system by definition of an extended constraint system. □

Lemma 10. *If $D \mapsto_{\mathsf{A/F}} D'$, then:*

1. $\mathsf{lev}(x, \mathsf{Act}(D')) \leq \mathsf{lev}(x, \mathsf{Act}(D))$ *for every* $x \in var(\mathsf{Act}(D))$.
2. *Let* $M = var(\mathsf{Act}(D')) \smallsetminus var(\mathsf{Act}(D))$. *If* $M \neq \emptyset$ *then there exists* $x \in var(\mathsf{Act}(D))$ *such that* $\mathsf{lev}(z, \mathsf{Act}(D')) < \mathsf{lev}(x, \mathsf{Act}(D))$ *for any* $z \in M \cup \{x\}$.

Proof. We distinguish several cases depending on the transformation rules involved in the step $D \mapsto_{\mathsf{A,F}} D'$.

Rule Triv or Comp (Active): In such a case, $var(\mathsf{Act}(D')) \smallsetminus var(\mathsf{Act}(D)) = \emptyset$ and it is easy to see that $\mathsf{lev}(x, \mathsf{Act}(D')) \leq \mathsf{lev}(x, \mathsf{Act}(D))$ for every $x \in var(\mathsf{Act}(D))$. Thus the result trivially holds.

Rule Axiom (Active): In such a case, $M = var(\mathsf{Act}(D')) \smallsetminus var(\mathsf{Act}(D)) = \emptyset$ since an Axiom rule does not introduce any new variable. Actually, we have that $\mathsf{Act}(D') = (\mathsf{Act}(D) \smallsetminus \{H \Vdash u\})\theta$ where $\theta = \mathsf{mgu}(u, v)$ with $v \in H$. The application of a substitution like θ can only decrease the level of the existing variables. The fact that we remove a constraint can not increase the level of a variable. This allows us to conclude.

Rule Dec (Freeze): In such a case, $M = var(\mathsf{Act}(D')) \smallsetminus var(\mathsf{Act}(D)) = \emptyset$. Thus we only have to show the first item. Actually it is easy to see that $\mathsf{lev}(x, \mathsf{Act}(D')) = \mathsf{lev}(x, \mathsf{Act}(D))$ for every $x \in var(\mathsf{Act}(D))$. Thus the result trivially holds.

Rule Dec (Active): Let $H \Vdash v$ be the constraint on which the Dec rule is applied. Let $\mathsf{R} = \dfrac{v_1 \ \ldots \ v_m \ w_1 \ \ldots \ w_n}{w}$ be the inference rule involved in this step. Let $x_0 \in var(v)$ such that $\mathsf{lev}(x_0, \mathsf{Act}(D)) = \mathsf{lev}(H, \mathsf{Act}(D))$ and θ be the mgu computed by the transformation.

- First, we show that $\mathsf{lev}(x_0, \mathsf{Act}(D')) \leq \mathsf{lev}(x_0, \mathsf{Act}(D))$. If $x_0 \in dom(\theta)$ then $\mathsf{lev}(x_0, \mathsf{Act}(D')) = 0$ and this allows us to conclude. Otherwise, since $v\theta = w\theta$, there exists $z \in var(w)$ such that $x_0 \in var(z\theta)$. We have (thanks to our Condition 2b on versatile rules) that:

$$var(w) \subseteq var(v_1, \ldots, v_m, w_1, \ldots, w_n) \subseteq var(v_1, \ldots, v_m)$$

  Hence, there exists i such that $z \in var(v_i)$. Moreover, we know that $v_i\theta = u_i\theta$ for some $u_i \in st(H)$. By origination, we have that $H_y \subsetneq H$ for every $y \in var(u_i)$. Now, either there is an $u' \in st(u_i)$ such that $z\theta = u'\theta$, or else there is a variable $y \in var(u_i)$ such that $z\theta \in st(y\theta)$. In both cases, we deduce that there exists $y_0 \in var(H)$ such that $x_0 \in var(y_0\theta)$. We deduce that

$$\mathsf{lev}(x_0, \mathsf{Act}(D')) \leq \mathsf{lev}(y_0, \mathsf{Act}(D)) < \mathsf{lev}(x_0, \mathsf{Act}(D)).$$

- We have that $\mathsf{lev}(x, \mathsf{Act}(D')) = \mathsf{lev}(x, \mathsf{Act}(D))$ for every $x \in var(\mathsf{Act}(D))$. Indeed, the application of a substitution can only decrease the level of a variable already in $\mathsf{Act}(D)$. If the variable disappear, its level becomes 0.
- Let $M = var(\mathsf{Act}(D')) \smallsetminus var(\mathsf{Act}(D))$. In order to conclude, it remains to show that $\mathsf{lev}(z, \mathsf{Act}(D')) < \mathsf{lev}(x_0, \mathsf{Act}(D))$ for any $z \in M$. As seen above, for each $z \in M$, there exists i such that $z \in var(v_i)$ and we have that $v_i\theta = u_i\theta$ for some $u_i \in st(H)$. Now, by origination, we have that $H_y \subsetneq H$ for every $y \in var(u_i)$. Now, either there is an $u' \in st(u_i)$ such that $z\theta = u'\theta$, or else there is a variable $y \in var(u_i)$ such that $z\theta \in st(y\theta)$. Since $z \in M$, we have that $z \notin dom(\theta)$. Thus, in both cases, we deduce that there exists $y_0 \in$

$var(H)$ such that $z \in var(y_0\theta)$. Hence, we conclude thanks to origination that

$$\mathsf{lev}(z, \mathsf{Act}(D')) \leq \mathsf{lev}(y_0, \mathsf{Act}(D)) < \mathsf{lev}(x_0, \mathsf{Act}(D)).$$

□

Lemma 11. *There is no infinite transformation sequence without opening the fridge.*

Proof. Given an extended constraint system D, we consider the multiset of the active levels of the variables that occur in $\mathsf{Act}(D)$, i.e.

$$M(D) = \{\mathsf{lev}(x, \mathsf{Act}(D)) \mid x \in var(\mathsf{Act}(D))\}.$$

These multisets are ordered according to the multiset extension of the ordering on natural numbers. Thanks to Lemma 10, we have $D \mapsto_{\mathsf{A},\mathsf{F}} D'$ implies that:

- either $var(\mathsf{Act}(D')) \subseteq var(\mathsf{Act}(D))$ and $M(D') \leq M(D)$;
- or else $M(D') < M(D)$.

Therefore, by the well-foundedness of the ordering on multisets of natural numbers, there can only be a finite number of rules that introduce variables, in any transformation sequence. Hence, in any infinite transformation sequence, there must be an infinite sequence of extended constraint systems whose set of variables in the active part is constant.

Let us show that, once the set of variables in the active part is constant, we can not add any new subterm in the active part of the constraint system. Indeed, in such a situation, an Active Dec is the only rule that may add new subterms to the active part. However, for each maximal new subterm $v_i\theta$ introduced by Active Dec, there is a corresponding $u_i \in st(\mathsf{Act}(D))$, such that $u_i\theta = v_i\theta$. Since the set of variables is fixed in the active part, we have $u_i\theta = u_i$ and therefore $v_i\theta = u_i \in st(\mathsf{Act}(D))$. Since any non-maximal $w_j\theta$ introduced by Dec is subterm of some maximal $v_i\theta$ (thanks to our Condition 2b), we conclude that the subterms of the constraints in the active part are fixed. Hence, there are only finitely many possible constraints systems: any infinite sequence must loop on the active part, which is forbidden by the strategy. □

Lemma 19. *Let $\mathsf{E} = \mathsf{E_A} \cup \mathsf{E_F}$ be a set of equations and $L = L_1 \uplus L_2$ be a set of variables that satisfy the following conditions:*

1. *$\mathsf{E_A}$ is a set of equations such that $\mathsf{mgu}(\mathsf{E_A}) \neq \perp$. We denote by $\sigma_\mathsf{A} = \mathsf{mgu}(\mathsf{E_A})$.*
2. *$L \cap var(\mathsf{E_F}) = \emptyset$*
3. *$\forall x \in L.\ \forall y \in dom(\sigma_\mathsf{A}).\ x \in var(y\sigma_\mathsf{A}) \implies y \in L_2$*
4. *$\forall x \in L_1.\ x \notin dom(\sigma_\mathsf{A})$*

Let $\theta = \mathsf{mgu}(\mathsf{E})$. Then we have that:

– $\forall x \in L_1. \forall y \in dom(\theta).\ x \in var(y\theta) \implies y \in L_2$;
– $\forall x \in L_1.\ x \notin dom(\theta)$.

Proof. Let $\sigma_{\mathsf{A}} = \sigma' \cup \sigma_0$ where $dom(\sigma') = dom(\sigma_{\mathsf{A}}) \cap L_2$. Thus, we have that $dom(\sigma_0) = dom(\sigma_{\mathsf{A}}) \smallsetminus L_2$ and thanks to point 4, we have that $dom(\sigma_0) \cap L_1 = \emptyset$. Hence, we have that $dom(\sigma_0) \cap L = \emptyset$. Now, thanks to point 3, we deduce that $var(img(\sigma_0)) \cap L = \emptyset$. Altogether, we conclude that $var(\mathsf{E_F} \cup \sigma_0) \cap L = \emptyset$.

Hence, we deduce that $\theta = \mathsf{mgu}(\mathsf{E_F} \cup \sigma_0 \cup \sigma') = \sigma' \mathsf{mgu}(\mathsf{E_F} \cup \sigma_0) \cup \mathsf{mgu}(\mathsf{E_F} \cup \sigma_0)$. Thus, we have that $dom(\theta) \cap L_1 = \emptyset$ and if $x \in var(y\theta) \cap L_1$ for some $y \in dom(\theta)$ then $y \in L_2$. This allows us to conclude. $\square$

Definition 9 (unsolved active level). *Let D be an extended constraint system. Its* unsolved active level *is its minimal level $\ell \in \{0, 1, \ldots, |\mathsf{LH}(\mathsf{Act}(D))|\}$ such that:*

for any constraint $H \Vdash v \in \mathsf{Act}(D)$ of level (strictly) greater than ℓ, we have that v is a variable and $\mathsf{lev}(v, \mathsf{Act}(D)) = \mathsf{lev}(H \Vdash v, \mathsf{Act}(D))$.

We will denote it by $\mathsf{unsolvedActlev}(D)$.

Intuitively, the set of constraints of level (strictly) greater than its unsolved active level ℓ contains exactly one constraint $H_x \Vdash x$, for each x with $\mathsf{lev}(x, \mathsf{Act}(D)) > \ell$, and no other constraints. Note that, since no rules but Triv apply to constraints of the form $H \Vdash x$, there is no transformation rule that can affect constraints whose level is greater than this level ℓ. Thus, the unsolved level of a constraint system is never increasing along a transformation sequence.

For an extended constraint system D, we denote by $\mathsf{sActlev}(x, D)$ the size (i.e. number of elements) of the left-hand side of the constraint that introduces the variable x in $\mathsf{Act}(D)$. By convention, $\mathsf{sActlev}(x, D) = 0$ when $x \notin \mathsf{Act}(D)$. Lastly, we denote by $\mathsf{sunsolvedActlev}(D)$ the size (i.e. number of elements) of the left-hand side of the constraints of level $\mathsf{unsolvedActlev}(D)$.

Lemma 20. *Let D, D' be two extended constraint systems such that $D \mapsto_{\mathsf{A/F}} D'$. For every $y \in var(D)$, we have that $\mathsf{sActlev}(y, D') \leq \mathsf{sActlev}(y, D)$.*

Proof. Let $y \in var(D)$. If $y \notin var(\mathsf{Act}(D))$ then $y \notin var(\mathsf{Act}(D'))$ and thus, we have that $\mathsf{sActlev}(y, D') = \mathsf{sActlev}(y, D) = 0$. Otherwise, we have that $y \in var(\mathsf{Act}(D))$. Let $H \Vdash v$ be the constraint in $\mathsf{Act}(D)$ that introduces y. We have that $y \in var(v)$ and $\mathsf{sActlev}(y, D) = |H|$. Now, we distinguish two cases depending on the fact that the transformation rule involved in $D \mapsto_{\mathsf{A/F}} D'$ is applied on $H \Vdash v$ or not.

– *The transformation rule is not applied on $H \Vdash v$.* In such a case, it may happen that $y \notin var(\mathsf{Act}(D'))$. This happens if we apply an Active rule and $y \in dom(\theta)$ where θ is the substitution involved in the application of the rule. In such a case, the result trivially holds. Otherwise, we have that $\mathsf{sActlev}(y, D') \leq \mathsf{sActlev}(y, D)$.

– *The transformation rule is applied on $H \Vdash v$.* Note that in such a case, this can not be a Freeze rule. Again, either we have that $y \notin \mathit{var}(\mathsf{Act}(D'))$ or otherwise, $\mathsf{sActlev}(y, D') \leq \mathsf{sActlev}(y, D)$. □

Lemma 12. *Let D be an extended constraint system such that $\mathsf{E}(D) = \mathsf{Fr}(D) = \emptyset$. In any transformation sequence starting with D, we can open at most $2|\mathsf{LH}(D)|+|\mathsf{MLH}(D)|$ times the fridge – $\mathsf{MLH}(D)$ is the maximal set of hypotheses in $\mathsf{LH}(D)$.*

Proof. We consider a transformation sequence as follows:

$$D_1 \mapsto^*_{\mathsf{A/F}} S_1 \mapsto_{\mathsf{O}} D_2 \mapsto^*_{\mathsf{A/F}} S_2 \mapsto_{\mathsf{O}} \cdots$$

Note that the constraint systems D_i have an empty frozen part whereas the constraint systems S_i have a solved active part. The remaining of the proof is to show that:

– $\mathsf{unsolvedActlev}(D_{i+1}) < \mathsf{unsolvedActlev}(D_i)$ or
– $|\mathsf{MLH}(D_{i+1})| < |\mathsf{MLH}(D_i)|$ or
– $|\mathsf{LH}(D_{i+1})| < |\mathsf{LH}(D_i)|$.

Since these three measures can not increase, this allows us to conclude that we can open at most $2|\mathsf{LH}(D)| + |\mathsf{MLH}(D)|$ times the fridge.

Assume that $|\mathsf{MLH}(D_{i+1})| = |\mathsf{MLH}(D_i)|$ and $|\mathsf{LH}(D_{i+1})| = |\mathsf{LH}(D_i)|$. Let ℓ be the unsolved active level of D_i and $k_\ell = \mathsf{sunsolvedActlev}(D_i)$. We will prove by induction on the length of the transformation sequence $D_i \mapsto^*_{\mathsf{A/F}} D' \mapsto_{\mathsf{A/F}} S$ several claims. We denote by $H \Vdash v$ the constraint in $\mathsf{Act}(D')$ to which the last transformation rule is applied and by θ the mgu that has been computed in this application. Note that θ can either be applied to the active part or simply stored in case of a Freeze rule.

Claim 1 *We have that* $\mathsf{sunsolvedActlev}(S) \leq k_\ell$.

The base case is obvious. Indeed, in such a case we have that $S = D_i$ and $\mathsf{sunsolvedActlev}(S) = k_\ell$. Now, assume by induction that $\mathsf{sunsolvedActlev}(D') \leq k_\ell$. Then, for any constraint $H_u \Vdash u \in \mathsf{Act}(D')$ with $|H_u|$ strictly greater than k_ℓ, we have that $\mathsf{lev}(H_u \Vdash u, \mathsf{Act}(D')) > \mathsf{unsolvedActlev}(D')$. Thus, u is a variable $H \subsetneq H_u$ and u can not appear in the domain or the image of θ.

In case of an Active rule, we deduce that $H_u\theta \Vdash u$ is the only constraint in $\mathsf{Act}(S)$ that contains the variable u in its right hand side among all the constraint in $\mathsf{Act}(S)$. In case of a Freeze rule, we deduce that $H_u \Vdash u$ is the only constraint in $\mathsf{Act}(S)$ that contains the variable u in the right hand side among all the constraint in $\mathsf{Act}(S)$. We have that $\mathsf{sunsolvedActlev}(S) \leq k_\ell$. ◇

Claim 2 *For every $y \in \mathit{var}(\mathsf{Fr}(S))$, we have that* $\mathsf{sActlev}(y, S) < k_\ell$.

The base case is obvious. Indeed, in such a case, we have that $S = D_i$ and $\mathsf{Fr}(S) = \emptyset$. Otherwise, let $y \in \mathit{var}(\mathsf{Fr}(S))$. We distinguish several cases:

– If $y \in var(\mathsf{Fr}(D'))$, we have by induction hypothesis that $\mathsf{sActlev}(y, D') < k_\ell$. By Lemma 20, we conclude $\mathsf{sActlev}(y, S) \leq \mathsf{sActlev}(y, D') < k_\ell$.
– Otherwise, a Freeze rule has been applied to $H \Vdash v$ and the variable y has been introduced in the frozen part of the constraint system. If y is a fresh variable, then $y \notin \mathsf{Act}(S)$ and $\mathsf{sActlev}(y, S) = 0$. We easily conclude. Otherwise, we have that $y \in var(H)$. By Claim 1 applied to D', we have $\mathsf{sunsolvedActlev}(D') \leq k_\ell$. Since a rule is applied to $H \Vdash v$, we have $|H| \leq \mathsf{sunsolvedActlev}(D')$. Since $y \in var(H)$, by origination, we have that $\mathsf{sActlev}(y, D') < |H|$ and hence $\mathsf{sActlev}(y, D') < k_\ell$. Therefore, by Lemma 20, we conclude that $\mathsf{sActlev}(y, S) < k_\ell$. $\diamond$

The lemma below will be useful to know the level of the different variables when we open the fridge.

Claim 3 *Assume that $mgu(\mathsf{E}(S)) \neq \bot$ and let $\sigma_S = mgu(\mathsf{E}(S))$. For all $x \in var(\mathsf{Act}(S))$, with $\mathsf{sActlev}(x, S) \geq k_\ell$, we have that:*

1. $x \notin dom(\sigma_S)$ and
2. $\forall y \in var(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) \cap dom(\sigma_S).\ x \notin var(y\sigma_S)$

For easing the proof, we split the set of equations $\mathsf{E}(S)$ in two parts:

– those that come from the application of an Active rule or from $\mathsf{E}(D_i)$, and
– those that come from the application of a Freeze rule.

We let $\mathsf{E}(S) = \mathsf{E}_\mathsf{A}(S) \cup \mathsf{E}_\mathsf{F}(S)$, with $\mathsf{E}(D_i) \subseteq \mathsf{E}_\mathsf{A}(S)$. We denote by σ_A the substitution $mgu(\mathsf{E}_\mathsf{A}(S))$ and by σ_i the substitution $mgu(\mathsf{E}(D_i))$. Let

– $L_1 = \{x \in var(\mathsf{Act}(S)) \mid \mathsf{sActlev}(x, S) \geq k_\ell\}$, and
– $L_2 = dom(\sigma_i) \cup \{y \mid \exists S', S''.\ D_i \mapsto^*_{\mathsf{A/F}} S' \mapsto_\mathsf{A} S'' \mapsto^*_{\mathsf{A/F}} S.\ \mathsf{sActlev}(y, S')) \geq k_\ell$ and $\mathsf{sActlev}(y, S'') = 0\}$.

Intuitively, the variables in L_1 are those that are introduced by a constraint whose size is greater than k_ℓ whereas those in L_2 are those that already disappeared by the application of the open rule on D_i or that disappear (by applying an Active rule) along the transformation from D_i to S.

Let us show that $\mathsf{E}(S) = \mathsf{E}_\mathsf{A}(S) \cup \mathsf{E}_\mathsf{F}(S)$ and $L = L_1 \uplus L_2$ satisfy the conditions required to apply Lemma 19.

1. $\mathsf{E}_\mathsf{A}(S)$ is a set of equations and $\mathsf{mgu}(\mathsf{E}_\mathsf{A}(S)) \neq \bot$ since $mgu(\mathsf{E}(S)) \neq \bot$.
2. $L \cap var(\mathsf{E}_\mathsf{F}(S)) = \emptyset$. By absurd, assume that there exists $y \in L \cap var(\mathsf{E}_\mathsf{F}(S))$.
   – If $y \in var(\mathsf{E}_\mathsf{F}(S))$, this means that a Freeze rule has been applied to a constraint $H' \Vdash v'$ in $\mathsf{Act}(S')$ for some S' such that $D_i \mapsto^*_{\mathsf{A/F}} S' \mapsto^*_{\mathsf{A/F}} S$ with $y \in var(H', v')$. In such a case, by definition of a Freeze rule, we have that $0 \neq \mathsf{lev}(y, \mathsf{Act}(S')) < \mathsf{lev}(H', \mathsf{Act}(S')) \leq \mathsf{unsolvedActlev}(S')$, and thus we have that $0 \neq \mathsf{sActlev}(y, S') < \mathsf{sunsolvedActlev}(S') \leq k_\ell$ by relying on Claim 1 for the last inequality.

- If $y \in L$ then we have that $\mathsf{sActlev}(y, S'') \geq k_\ell$ or $\mathsf{sActlev}(y, S'') = 0$ for each S'' along the derivation $D_i \mapsto^*_{\mathsf{A/F}} S$. This is by definition of L_1, L_2 and by Lemma 20.

Thus, in one hand we have that $0 \neq \mathsf{sActlev}(y, S') < k_\ell$, and in the other hand, we have that $\mathsf{sActlev}(y, S') \geq k_\ell$ or $\mathsf{sActlev}(y, S') = 0$. This is impossible. Hence, we conclude that $L \cap \mathit{var}(\mathsf{E_F}(S)) = \emptyset$.

3. Let $x \in L$ and consider the set Y_x defined as follows:

$$Y_x = \{y \in \mathit{dom}(\sigma_{\mathsf{A}}) \mid y \notin L_2 \text{ and } x \in \mathit{var}(y\sigma_{\mathsf{A}})\}$$

Assume by absurd that Y_x is not empty. Let $y \in Y_x$ be such that $y\sigma_{\mathsf{A}}$ is minimal among $\{z\sigma \mid z \in Y_x\}$ w.r.t. the subterm ordering. Since $y \in \mathit{dom}(\sigma_{\mathsf{A}})$ and $y \notin L_2$, we have that:

- there are two constraint systems S', S'', with $D_i \mapsto^*_{\mathsf{A/F}} S' \mapsto_{\mathsf{A}} S'' \mapsto^*_{\mathsf{A/F}} S$ such that $y \in \mathit{var}(\mathsf{Act}(S'))$ and $y \notin \mathit{var}(\mathsf{Act}(S''))$; and
- $0 \neq \mathsf{sActlev}(y, S') < k_\ell$. We consider the constraint in $H' \Vdash v' \in \mathsf{Act}(S')$ witnessing this fact, i.e. $y \in \mathit{var}(v')$ and $\mathsf{lev}(H', \mathsf{Act}(S')) = \mathsf{lev}(y, \mathsf{Act}(S'))$ and $|H'| < k_\ell$.

Let θ be the mgu computed in the application of $S' \mapsto_{\mathsf{A}} S''$ and $H \Vdash v$ be the constraint to which the rule is applied.

Note that $x \in \mathit{var}(y\theta)$. Indeed, if this is not the case, this means that there is an $y' \in \mathit{var}(y\theta)$ such that $y' \in \mathit{var}(\mathsf{Act}(S''))$ and $x \in \mathit{var}(y'\sigma_{\mathsf{A}})$. To contradict the minimality of y, we have to show that $y' \in Y_x$. To establish this, we will first show that $\mathsf{sActlev}(y', S'') < k_\ell$. We distinguish two cases:

(a) If $\mathsf{sActlev}(y', S') < k_\ell$, we easily conclude by Lemma 20.

(b) Otherwise we have that $\mathsf{sActlev}(y', S') \geq k_\ell$. Since $y' \in \mathit{img}(\theta)$, by Claim 1, we deduce that $\mathsf{sActlev}(y', S') = k_\ell$ and hence $|H| = k_\ell$. Since $|H'| < k_\ell$, we deduce that $H' \subsetneq H$. Therefore, we have that $H'\theta \Vdash v'\theta \in \mathsf{Act}(S')$ and hence $\mathsf{sActlev}(y', S'') < k_\ell$.

Hence, in both cases, we have that $\mathsf{sActlev}(y', S'') < k_\ell$ and thanks to Lemma 20, we know that the active level of the variable y' decreases along the derivation. Thus, by definition of the set L_2, we have that $y' \notin L_2$ (note that $y' \notin \mathit{dom}(\sigma_i)$). Hence, we contradict the minimality of y. We conclude that $x \in \mathit{var}(y\theta)$.

Now, we distinguish two cases:

(a) Either $H \Vdash v = H' \Vdash v'$. In other words, the constraint on which the transformation rule is applied to go from S' to S'' is the constraint witnessing the fact that $\mathsf{sActlev}(y, S') < k_\ell$. Since $x \in \mathit{var}(y\theta)$, we know that x occurs in $H \Vdash v$, and thus $0 \neq \mathsf{sActlev}(x, S') \leq |H| = |H'| < k_\ell$.

(b) Otherwise, we have that $H \Vdash v \neq H' \Vdash v'$. In such a case, we have that $H'\theta \Vdash v'\theta \in \mathsf{Act}(S'')$ and $|H'\theta| < k_\ell$. Since, we have that $x \in \mathit{var}(y\theta)$ and $y \in \mathit{var}(v')$, we easily deduce that $0 \neq \mathsf{sActlev}(x, S'') < k_\ell$.

By applying Lemma 20, in both cases, we deduce that $\mathsf{sActlev}(x, S) < k_\ell$. Hence, we have that $x \notin L_1$. Since x occurs in S', it is also clear that $x \notin \mathit{dom}(\sigma_i)$. Now, by Lemma 20, the size of the active level of x can only decreases along the derivation from D_i to S. Since, this size was different from 0 but less than k_ℓ at some point in the derivation, we deduce that $x \notin L_2$. Hence, we have that $x \notin L$. Contradiction.

4. $L_1 \cap dom(\sigma_{\mathsf{A}}) = \emptyset$. Indeed, we have that $L_1 \subseteq var(\mathsf{Act}(S))$ and also that $dom(\sigma_{\mathsf{A}}) \cap var(\mathsf{Act}(S)) = \emptyset$ (the Active equations have been applied to the active part).

Hence, we may apply Lemma 19. Since $\sigma_S = \mathsf{mgu}(\mathsf{E}(S))$, Lemma 19 tells us that for all $x \in var(\mathsf{Act}(S))$ with $\mathsf{sActlev}(x, S) \geq k_\ell$,

1. $x \notin dom(\sigma_S)$
2. $x \in var(y\sigma_S)$, with $y \in dom(\sigma_S)$, implies $y \in L_2$. To conclude, it is now sufficient to show that $L_2 \cap var(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) = \emptyset$.
   - The fact that $L_2 \cap var(\mathsf{Act}(S)) = \emptyset$ follows immediately from the definition of L_2.
   - Let $y \in L_2$. If $y \in dom(\sigma_i)$, then we easily conclude. Otherwise, let S', S'' be the constraint systems witnessing the fact that $y \in L_2$. We have that $\mathsf{sActlev}(y, S') \geq k_\ell$ and by Claim 2, we deduce that $y \notin var(\mathsf{Fr}(S'))$. Since $S' \mapsto_{\mathsf{A}} S''$ (by application of an Active rule), we deduce that $y \notin var(\mathsf{Fr}(S''))$. Actually, since $\mathsf{sActlev}(y, S'') = 0$ (which means that y does not occur in the Active part), there is no way to put y in the frozen part. Thus, for all $S'' \mapsto^* S'''$ we have that $y \notin var(\mathsf{Fr}(S'''))$. Hence, we deduce that $y \notin var(\mathsf{Fr}(S))$.

   Then we conclude $\forall y \in var(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) \cap dom(\sigma_S).\ x \notin var(y\sigma_S)$. $\diamond$

We let $\mathsf{LH}(\mathsf{Fr}(S))$ be the left-hand sides of constraints in $\mathsf{Fr}(S)$.

Claim 4 *For all $H' \in \mathsf{LH}(\mathsf{Fr}(S))$, we have that $|H'| < k_\ell$.*

Again the base case is obvious since $S = D_i$ and thus $\mathsf{Fr}(S) = \emptyset$. Now, let $H' \in \mathsf{LH}(\mathsf{Fr}(S))$. If $H' \in \mathsf{LH}(\mathsf{Fr}(D'))$, we deduce immediately by induction hypothesis that $|H'| < k_\ell$. Otherwise, if $H' \in \mathsf{Fr}(S) \smallsetminus \mathsf{Fr}(D')$, this means that a Freeze rule has been applied. By definition of Freeze, we have $H' \subsetneq H$ and thus $|H'| < |H|$. Since $|H| \leq \mathsf{sunsolvedActlev}(D')$ and, by Claim 1, we have that $\mathsf{sunsolvedActlev}(D') \leq k_\ell$, we conclude that $|H'| < k_\ell$. $\diamond$

We conclude that the four claims hold for $S = S_i$.

Assume that $mgu(\mathsf{E}(S_i)) \neq \bot$ and let $\sigma_i = mgu(\mathsf{E}(S_i))$. We have that:

$$D_{i+1} = \exists \tilde{z}.[(\mathsf{Act}(S_i) \cup \mathsf{Fr}(S_i))\sigma_i \mid \emptyset \mid \sigma_i].$$

Let $H \Vdash u \in (\mathsf{Act}(S_i) \cup \mathsf{Fr}(S_i))\sigma_i$ be a constraint of $\mathsf{Act}(D_{i+1})$ such that $\mathsf{lev}(H, \mathsf{Act}(D_{i+1})) \geq \ell$. First, since $|\mathsf{MLH}(D_{i+1})| = |\mathsf{MLH}(D_i)|$ and $|\mathsf{LH}(D_{i+1})| = |\mathsf{LH}(D_i)|$, we have $|H| \geq k_\ell$. Thus, by Claim 4, we have $H \Vdash u \notin \mathsf{Fr}(S_i)\sigma_i$. Then we must have $H \Vdash u \in \mathsf{Act}(S_i)\sigma_i$. Since $\mathsf{Act}(S_i)$ is in solved form, there is a $H' \Vdash x \in \mathsf{Act}(S_i)$, with $(H' \Vdash x)\sigma_i = H \Vdash u$ and $|H'| = |H| \geq k_\ell$ (we have again used $\mathsf{MLH}(D_{i+1}) = \mathsf{MLH}(D_i)$). Then $\mathsf{sActlev}(x, S_i) \geq k_\ell$ and, by Claim 3 (point 1), we have that $x \notin dom(\sigma_i)$. We deduce that $u = x$. By point 2 of Claim 3, we have that

$$\forall y \in var(\mathsf{Act}(S_i) \cup \mathsf{Fr}(S_i)) \cap dom(\sigma_i).\ x \notin var(y\sigma_i).$$

Moreover, by Claim 2, we have that $x \notin \mathsf{Fr}(S_i)$. Thus, we have that this occurrence of x is the only occurrence of x in the right hand side of the deducibility constraints of D_{i+1}. Hence we conclude that the unsolved level of D_{i+1} is strictly smaller than ℓ and we are done. □

Corollary 2 (of Lemma 7). *Let D be a uniquely determined extended constraint system such that $\mathsf{Act}(D)$ is not in solved form and $\sigma \in Sol(D)$. Then there is a D' such that $D \mapsto_{\mathsf{A/F}} D'$ and $\sigma \in Sol(D')$.*

Moreover, $\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{var(\mathsf{A})}) > \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{var(\mathsf{A}')})$ where $\mathsf{A} = \mathsf{Act}(D)$, $\mathsf{A}' = \mathsf{Act}(D')$ and $\overline{\sigma}$ is the extension of σ w.r.t. D'.

Proof. Let $D = \exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$ be an extended constraint system that is uniquely determined and $\sigma \in Sol(D)$. Let σ' be the extension of σ w.r.t. D. Note that $\sigma'|_{var(\mathsf{A})}$ is a solution of A and A is not in solved form. By Lemma 7, we get $[\mathsf{A} \mid \emptyset] \rightsquigarrow \exists\tilde{z'}.[\mathsf{A}' \mid \theta]$, such that $\sigma'|_{var(\mathsf{A})} \in Sol(\exists\tilde{z'}.[\mathsf{A}' \mid \theta])$ and $\mathsf{PS}(\mathsf{A}, \sigma'|_{var(\mathsf{A})}) > \mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \sigma'|_{var(\mathsf{A})})$. By the definition of the transformation rules for extended constraint systems, we deduce that:

- either we can apply an Active rule, to get $D \mapsto_{\mathsf{A}} D' = \exists\tilde{z} \cup \tilde{z'}.[\mathsf{A}' \mid \mathsf{F} \mid \mathsf{E} \cup \theta]$. We have that D' is an extended constraint system such that $\mathsf{Act}(D') = \mathsf{A}'$. Obviously, we have that $\sigma \in Sol(D')$. D' is uniquely determined and $\overline{\sigma} = \sigma' \cup \sigma_0$, where $dom(\sigma_0) = \tilde{z'}$.
  Moreover, we have $\mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \sigma'|_{var(\mathsf{A})}) = \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{var(\mathsf{A}')})$. Therefore, we have that

  $$\mathsf{PS}(\mathsf{A}, \sigma'|_{var(\mathsf{A})}) > \mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \sigma'|_{var(\mathsf{A})}) = \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{var(\mathsf{A}')}).$$

  Since $\overline{\sigma}|_{var(\mathsf{A})} = \sigma'|_{var(\mathsf{A})}$, this allows us to conclude that $\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{var(\mathsf{A})}) > \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{var(\mathsf{A}')})$ where $\mathsf{A} = \mathsf{Act}(D)$ and $\mathsf{A}' = \mathsf{Act}(D')$.
- or else we can apply a Freeze rule, to get $D \mapsto_{\mathsf{F}} D' = \exists\tilde{z} \cup \tilde{z'}.[\mathsf{A}_a \mid \mathsf{F} \cup \mathsf{A}_f \mid \mathsf{E} \cup \theta]$, where $\mathsf{A}' = (\mathsf{A}_a \cup \mathsf{A}_f)\theta$. Again, it is easy to see that $\sigma \in Sol(D')$. Moreover, we have that $\mathsf{Act}(D') = \mathsf{A}_a$ and $\mathsf{A}_a\overline{\sigma} \subseteq \mathsf{A}'\overline{\sigma}$. Thus, we have that:

  $$\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{var(\mathsf{A})}) > \mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \overline{\sigma}|_{var(\mathsf{A})}) \geq \mathsf{PS}(\mathsf{A}_a, \overline{\sigma}|_{var(\mathsf{A}_a)}).$$

□

Lemma 13. *Our strategy is complete.*

Proof. Let D_0 be an extended constraint system and $\sigma \in Sol(D_0)$. We have to show that there exists D' in solved form such that $\sigma \in Sol(D')$ and $D_0 \mapsto^* D'$ by a sequence authorized by the strategy. Thanks to Corollary 2, we know that there is a sequence

$$D_0 \mapsto^*_{\mathsf{A/F}} S_1 \mapsto_{\mathsf{O}} D_1 \mapsto^*_{\mathsf{A/F}} S_2 \mapsto_{\mathsf{O}} D_2 \mapsto^*_{\mathsf{A/F}} \dots$$

such that σ is a solution of each constraint system in the sequence and such that the measure PS on the active part decreases along each sequence $D_i \mapsto^*_{\mathsf{A/F}} S_{i+1}$.

Moreover, thanks to Corollary 1, we know that we will close this sequence at some point. Let D' be the last constraint system of this sequence. To conclude, it remains to show that D' is in solved form. Assume by absurd that D' is not in solved form. We distinguish two cases:

- $\mathsf{Act}(D')$ is in solved form: Open can be applied leading to a contradiction.
- $\mathsf{Act}(D')$ is not in solved form: Thanks to Corollary 2, we deduce that there is an extended constraint system D'' such that $D' \mapsto_{\mathsf{A/F}} D''$, $\sigma \in Sol(D'')$ and $\mathsf{PS}(\mathsf{Act}(D'), \overline{\sigma}|_{var(\mathsf{Act}(D'))}) > \mathsf{PS}(\mathsf{Act}(D''), \overline{\sigma}|_{var(\mathsf{Act}(D''))})$, where $\overline{\sigma}$ is the extension of σ w.r.t. D''.
  By the fact that $D' \mapsto_{\mathsf{A/F}} D''$ is forbidden by the strategy, we deduce that there is (in the sequence we consider) a subsequence $D_j \mapsto^*_{\mathsf{A/F}} D'$ such that $\mathsf{Act}(D_j) = \mathsf{Act}(D'')$ (a looping branch is cut). This implies that our measure PS has not decreased during this subsequence. Thus, contradiction. $\square$

D An example

We give here a (trivial) example illustrating how we can get a good inference system by saturation of a non-good one.

Consider the system containing the single versatile rule

$$\frac{\mathsf{f}(\mathsf{g}(x), \mathsf{g}(y)) \quad \mathsf{g}(x) \quad y}{\mathsf{h}(x, y)}$$

and a composition rule for each of $\mathsf{f}, \mathsf{g}, \mathsf{h}$. This inference system satisfies our conditions on the form of rules, but it is not good, since there are only bad proofs of e.g. $\mathsf{g}(a), \mathsf{g}(b), b \vdash \mathsf{h}(a, b)$.

We can complete the system into a good one by saturation. First, overlap the only versatile rule and the rule

$$\frac{x_1 \quad x_2}{\mathsf{f}(x_1, x_2)}$$

We get a new versatile rule

$$\frac{\mathsf{g}(x) \quad \mathsf{g}(y) \quad y}{\mathsf{h}(x, y)}$$

that we add to our inference system. Now, we have a good proof of $\mathsf{g}(a), \mathsf{g}(b), b \vdash \mathsf{h}(a, b)$. However, the system is not good yet, since there are only bad proofs of e.g. $\mathsf{g}(a), b \vdash \mathsf{h}(a, b)$. Another ordered resolution step, overlapping

$$\frac{y}{\mathsf{g}(y)}$$

and the new versatile rule, adds the rule

$$\frac{\mathsf{g}(x) \quad y}{\mathsf{h}(x, y)}$$

The resulting system displayed below is good, since all bad proofs are now avoidable.

$$\frac{x \quad y}{\mathsf{f}(x, y)} \qquad \frac{x \quad y}{\mathsf{h}(x, y)} \qquad \frac{x}{\mathsf{g}(x)}$$

$$\frac{\mathsf{f}(\mathsf{g}(x), \mathsf{g}(y)) \quad \mathsf{g}(x) \quad y}{\mathsf{h}(x, y)} \qquad \frac{\mathsf{g}(x) \quad \mathsf{g}(y) \quad y}{\mathsf{h}(x, y)} \qquad \frac{\mathsf{g}(x) \quad y}{\mathsf{h}(x, y)}$$