Satisfiability of General Intruder Constraints with and without a Set Constructor

Tigran Avanesov, Yannick Chevalier, Michaël Rusinowitch, Mathieu Turuani

Satisfiability of General Intruder Constraints with and without a Set Constructor

Tigran Avanesov

SnT, Université du Luxembourg, Luxembourg

Yannick Chevalier

Université Paul Sabatier & IRIT Toulouse

Michael Rusinowitch

INRIA Nancy–Grand Est & LORIA, 54600 Villers-lès-Nancy, France

Mathieu Turuani

INRIA Nancy–Grand Est & LORIA, 54600 Villers-lès-Nancy, France

Abstract

Many decision problems on security protocols can be reduced to solving deduction constraints expressing whether an instance of a given message pattern can be constructed by the intruder. Most constraint solving procedures for protocol security rely on two properties of constraint systems called *monotonicity* and *variable-origination*. In this work we relax these restrictions by giving a decision procedure for solving general intruder constraints (that do not have these properties) that stays in NP. The result is also valid modulo an associative, commutative and idempotent theory. The procedure can be applied to verify security protocols in presence of multiple intruders.

Key words: ACI, Deducibility constraints, Dolev-Yao deduction system, Multiple intruders, Security

Email addresses: tigran.avanesov@uni.lu (Tigran Avanesov), yannick.chevalier@irit.fr (Yannick Chevalier), michael.rusinowitch@inria.fr (Michael Rusinowitch), mathieu.turuani@inria.fr (Mathieu Turuani).

[1] The work presented in this paper was partially supported by the FP7-ICT-2007-1 Project no. 216471,

1. Introduction

Detecting flaws in security protocol specifications under the perfect cryptography assumption in the Dolev-Yao intruder model is an approach that has been extensively investigated in recent years Armando et al. (2014); Groza and Minea (2013); Viganò (2012); Meadows (2011); Blanchet (2009); Guttman (2007); Arapinis and Duflot (2007); Turuani (2006). In particular, symbolic constraint solving has proved to be a very successful approach in the area. It amounts to expressing the possibility of mounting an attack, e.g. the derivation of a secret, as a list of steps where for each step an instance of the message pattern awaited according to the protocol has to be derived from the current intruder knowledge. These steps correspond in general to the progression of the protocol execution, up to the last one which is the secret derivation.

Enriching the standard Dolev-Yao intruder model with different equational theories Comon-Lundh and Shmatikov (2003); Comon-Lundh (2004); Basin et al. (2005); Chevalier and Rusinowitch (2010); Baskar et al. (2010); Escobar et al. (2011) like exclusive OR, modular exponentiation, Abelian groups, etc. Liu and Lynch (2011); Erbatur et al. (2011); Malladi (2012); Chevalier et al. (2005); Küsters and Truderung (2008) helps to find flaws that could not be detected considering free symbols only. A particularly useful theory is the theory of an ACI operator (that is associative, commutative and idempotent) since it allows one to express sets in cryptographic protocols.

Up to the exception of Mazaré (2005), all proposed algorithms rely on two strong assumptions about the constraints to be processed:

- knowledge monotonicity, reflecting the fact the the intruder could see everything that occurred before and forgot nothing;
- variable origination, reflecting that each variation in the protocol is introduced by the intruder.

Constraints satisfying these hypotheses are called *well-formed constraints* in the literature. Well-formed constraints are sufficient to solve security problems in the standard case where a single Dolev-Yao intruder is assumed. However, we will see that in some situations it can be quite useful to relax these hypotheses and consider *general constraints*, that is constraints without the restrictions above. General constraints naturally occur when considering security problems involving several non-communicating Dolev-Yao intruders (see § 2.1). Note that if intruders can communicate during protocol execution, the model becomes attack-equivalent to one with a unique Dolev-Yao intruder Syverson et al. (2000). A discussion on a multiple non-collaborating attackers model as well as interesting examples can be found in Fiazza et al. (2012).

1.1. Contributions of the paper

First, we will show that as for the standard case, in this more general framework it is still possible to derive an NP decision procedure for detecting attacks on a bounded number of protocol sessions (Sections 5, 4). Second, our result extends previous ones by allowing non-atomic keys (which is an important feature for protocol design as it is common to build symmetric keys from shared secrets) and the usage of an associative

"AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures" (`http://www.avantssar.eu`) and FP7-ICT Project no. 256980, "NESSoS: Network of Excellence on Engineering Secure Future Internet Software Services and Systems" (`http://www.nessos-project.eu`).

commutative idempotent operator (Sections 3, 4) that can be used for instance to model sets of nodes in XML document (see § 2.2). This extension of well-formed constraint systems may seem trivial but a third contribution of this paper is to demonstrate this is not the case by considering subterm deduction systems which are akin to the Dolev-Yao deduction system, but in which the equational theory can be any subterm convergent one. Whereas the decidability of well-formed constraint systems for subterm deduction systems is well known, see *e.g.* Baudet (2005), we prove in Appendix A that the satisfiability of general constraint systems is not decidable for subterm deduction systems. Finally we will sketch several applications of our results to security analysis in Section 2.

1.2. Related work

The decision procedure for satisfiability of well-formed constraint systems can be used to decide the insecurity of cryptographic protocols with a bounded number of sessions Rusinowitch and Turuani (2003). In this domain, several works deviated from the perfect cryptography assumption and started to consider algebraic properties of functional symbols. For example properties of XOR operator and exponentiation were considered in Lynch and Meadows (2004); Chevalier et al. (2005, 2008); Dougherty and Guttman (2013) and together with homomorphic symbol in Delaune (2006). A class of monoidal equational theories was studied in Delaune et al. (2008) including the ACUI theory which is quite similar to the ACI theory discussed in the present work, but the absence of a unit element in ACI seems to put the ACI theory out of scope of their method. Moreover Delaune et al. (2008) does not show the NP complexity of their procedure. We note that in Chevalier et al. (2007) the authors showed the *NP*-completeness for the AC theory, while Dolev-Yao deduction system was extended with a rule $x \cdot y \to x$, where $\cdot$ is an AC symbol. We will examine the ACI theory within the similar deduction system. Note also that some algebraic properties (like associative and commutative symbol) make the insecurity problem undecidable Bursuc et al. (2007).

All the decidability results mentioned above consider systems of constraints with two restrictions namely knowledge monotonicity (the left-hand side of a constraint representing the current knowledge of the intruder is included into the left-hand side of the next one) and variable origination (variable appears first in the right-hand side of some constraint): this limitation is not impeding the solution of usual protocol insecurity problems since the constraints generated with an active Dolev-Yao intruder are of the required type. An attempt to swerve from well-formed constraints was made in Mazaré (2005) where the knowledge monotonicity was relaxed to define "quasi well-formed" whose satisfiability is proven to be in NP. This result was later extended in Mazaré (2006) to general constraint systems but under the provision that all keys are atomic, an hypothesis that permits to guess at the start of the procedure which keys are available at which step to the intruder. However to our knowledge no extension of Dolev-Yao deduction system to non-atomic symmetric key or to algebraic properties has been shown decidable for general constraint systems. Moreover, satisfiability of well-formed constraints with the ACI theory was not considered before. We have presented our results without proofs in a conference paper Avanesov et al. (2010). In this long version we give detailed proofs. We believe that they can be useful for readers that want to study general constraints for other intruder theories.

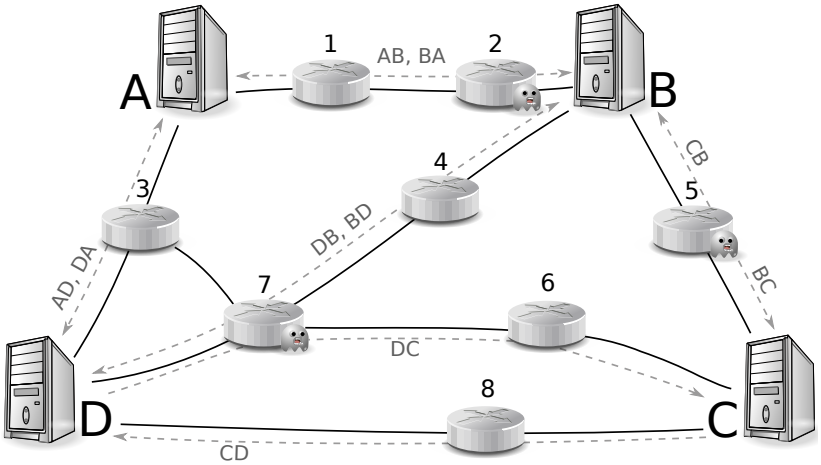


Figure 1. Untrusted routers

2. Motivating examples

2.1. Protocol analysis with several intruders

Security analysis of a protocol is usually conducted in the worst possible case: a single *intruder* (or a coalition of several intruders, but able to share information faster than normal commnications) controls all the network. However some protocols, especially routing protocols, are designed to work through different, non-communicating sub-networks. A relevant analysis of such protocols must take into account that the security of the protocol depends on the (correct) assumption that some communications are not possible. In these cases one needs to consider a less powerful intruder, or rather a collection thereof, that still controls some parts of the network but remains completely ignorant of what happens in other parts.

Suppose several agents $(A, B \dots$, see Figure 1) execute a protocol i.e. a specified finite sequence of message emissions or receptions. Due to their (long distance) layout they have to transmit data through routers $(1, 2, 3 \dots)$. The routing tables of all honest routers/agents are static (messages follow always the same path). Some routers $(2, 5, 7)$ may be compromised: an intruder managed to install a device controlling input and output of the router or implanted there his malicious code. A message circulated via such an untrusted channel (e.g. DB) is consumed by the corresponding compromised device (*local intruder*) (7) thereby increasing his knowledge. Moreover, a local intruder can forge and emit to an endpoint (C, B, D) of any channel he controls (BD, DB, DC) any message he can build using the content of his memory and some available transformations specified by a deduction system. Because of the network topology malicious routers have no means to communicate (there are no links between them, neither direct nor via other routers), but at some point the intruder can gather the knowledge of all the compromised routers (by physically collecting devices or reading their memory).

In this framework the *Coordinated attack problem* is to decide whether it is possible to initially give instructions to compromised routers (e.g. by reprogramming malicious devices) to force an execution such that honest agents following the protocol will reveal some secret data to the intruder (i.e. the intruder can deduce it using information collected at the end from all local intruders).

2.1.1. The coordinated attack problem

We give here a semi-formal description of the problem and some directions to solve it. A formal model can be found in Avanesov et al. (2010) or Avanesov (2011).

First we introduce some notations and definitions that will be detailed in the technical part of the paper (§ 3).

2.1.1.1. Messages. We consider messages as first-order terms built from a set of function symbols (such as encryption, pairing, etc.), a set of constants $\mathcal{A}$ (representing elementary pieces of data: texts, public keys, names of agents, etc) and a set of variables $\mathcal{X}$. Let $\mathcal{T}$ be the set of all possible terms. For a term t we denote by $\mathrm{Vars}(t)$ the set of all variables in t. A term t is a *ground term*, if $\mathrm{Vars}(t) = \emptyset$. The set of ground terms is denoted by $\mathcal{T}_g$.

We define a substitution $\sigma = \{x_1 \mapsto t_1, \ldots, x_k \mapsto t_k\}$ (where $x_i \in \mathcal{X}$ and $t_i \in \mathcal{T}$) to be the mapping $\sigma : \mathcal{T} \to \mathcal{T}$, such that $t\sigma$ is a term obtained by replacing, for all i, each occurrence of a variable x_i by the corresponding term t_i. If $T \subseteq \mathcal{T}$, then by definition $T\sigma = \{t\sigma : t \in T\}$. A substitution σ is *ground* if for any $i \in \{1, \ldots, k\}$, t_i is ground.

2.1.1.2. Agents. We will call honest communicating parties *agents*. Every agent is identified by its name. We denote the set of agent names by A.

2.1.1.3. Channels. Any two agents a and b communicate through a directed channel $a \rightharpoonup b$ implemented as a queue. The sent messages are stored in a queue to be processed in order of arrival.

2.1.1.4. Agents' behavior. Each agent has a specific sequence of actions to execute. Every action is either of reception type $?_f r$ (expecting a message r from agent f) or emission type $!_t s$ (sending a message s to an agent with name t). Here r and s are terms representing message patterns. For example, if the sequence of actions of agent a is $?_b \mathrm{aenc}(X, pk_a), !_c \mathrm{aenc}(X, pk_c)$ (where X is a variable and pk_a, pk_c are constants denoting a public key of a and a public key of c correspondingly and $\mathrm{aenc}(u, v)$ is an asymmetric encryption of u with key v) then a waits until he/she receives some message t matching a pattern $\mathrm{aenc}(X, pk_a)$ on a channel $b \rightharpoonup a$ and sends a message $\mathrm{aenc}(X\sigma, pk_b)$ on channel $a \rightharpoonup c$, where σ is a substitution satisfying $t = \mathrm{aenc}(X, pk_a)\sigma$. For instance, if a receives $\mathrm{aenc}(s, pk_a)$, i.e. some atomic value s asymmetrically encrypted with a public key of a pk_a, then the corresponding substitution σ is $\{X \mapsto s\}$ and a sends to c the value of s reencrypted with public key pk_c of c. On the other hand, it could be $\sigma = \{X \mapsto \mathrm{pair}(b, s)\}$ if b sent $\mathrm{aenc}(\mathrm{pair}(b, s), pk_a)$. Note that if the message does not match the expected pattern, the agent stops this execution. Once a variable is instantiated on a reception, its value can be used in the following agent actions.

2.1.1.5. Intruder model. We assume that some communication channels are controlled by N *local* intruders from a set $\mathbb{I} = \{I_i\}_{i=1,\ldots,N}$ and no channel is controlled by more than one intruder. Given some channel c, we denote by $\iota(c)$ the local intruder that controls c when there is one.

Every intruder I is given some initial knowledge K_I^0 that is a set of ground terms. Once an agent sends a message via a channel controlled by an intruder, the intruder reads it and blocks it. Reading the message means extending intruder's current knowledge with this message. An intruder controlling a channel can generate a message from his knowledge using deduction rules and send it to its endpoint.

We now specify the intruder capabilities.

Definition 2.1. A *rule* is a tuple of terms written as $s_1, \ldots, s_k \to s$, where $s_1, \ldots, s_k, s$ are terms. A *deduction system* $\mathcal{D}$ is a set of rules.

As an example of deduction system, we refer to a version of the classical Dolev-Yao deduction system (DY) presented in Table 1.

Table 1. DY deduction rules

Composition rules	Decomposition rules
$t_1, t_2 \to \text{enc}(t_1, t_2)$	$\text{enc}(t_1, t_2), t_2 \to t_1$
$t_1, t_2 \to \text{aenc}(t_1, t_2)$	$\text{aenc}(t_1, t_2), \text{priv}(t_2) \to t_1$
$t_1, t_2 \to \text{pair}(t_1, t_2)$	$\text{pair}(t_1, t_2) \to t_1$
$t_1, \text{priv}(t_2) \to \text{sig}(t_1, \text{priv}(t_2))$	$\text{pair}(t_1, t_2) \to t_2$

From now to the end of this section rules are assumed to belong to a fixed deduction system $\mathcal{D}$ and that terms are evaluated modulo a congruence on terms $\equiv_{\mathcal{E}}$ by an equational theory (a set of equations between terms together with the properties of equality) $\mathcal{E}$.

Given two sets of ground terms E, F and a rule $l \to r$, we write $E \to_{l \to r} F$ iff $F = E \cup \{s\}$, $s \equiv_{\mathcal{E}} r$ and $l \subseteq E$, where l is a set of terms. We write $E \to F$ iff there exists rule $l \to r$ such that $E \to_{l \to r} F$.

Definition 2.2. A *derivation* D is a finite sequence of finite sets of ground terms $E_0, E_1, \ldots, E_n$ such that $E_0 \to E_1 \to \cdots \to E_n$, where $E_i = E_{i-1} \cup \{t_i\}$, for all $i \in \{1, \ldots, n\}$. A term t is *derivable modulo an equational theory* $\mathcal{E}$ from a set of terms E iff there exists a derivation $D = E_0, \ldots, E_n$ such that $E_0 = E$ and there exists $s \in E_n$ such that $s \equiv_{\mathcal{E}} t$. We denote by $\text{der}_{\mathcal{E}}(E)$ the set of terms derivable from E modulo $\mathcal{E}$. A set of terms T is derivable from E, iff $T \subseteq \text{der}_{\mathcal{E}}(E)$.

When the equational theory $\mathcal{E}$ is clear from context we write $\text{Der}(E)$ instead of $\text{der}_{\mathcal{E}}(E)$, and similarly omit any reference to it. Informally, a local intruder may build new messages from the messages he knows by applying the deduction rules (e.g. concatenate two messages he knows). At each step a local intruder having recorded messages $t_1, \ldots, t_n$ can send a message m if $m \in \text{Der}(\{t_1, \ldots, t_n\})$.

2.1.1.6. Protocol execution. Given a set of protocol participants (agents) with their sequences of actions to execute, communication channels and local intruders each one with some initial knowledge, the course of events develops as follows. Each agent access channels connecting her to other agents. To send a message to Bob Alice inserts the message in the queue representing the channel, and to receive it Bob takes the first message in the queue and matches it against the pattern of the awaited message. An intruder Charlie controlling the channel may add, remove, read or reorder as he wishes messages in the queue.

2.1.1.7. Offline communication. At some execution point initially specified, the current knowledge of all local intruders is shared in order to derive a secret which probably they cannot deduce separately. We consider our intruder model is reasonable for covering situations where offline interactions between intruders are time-consuming and may be detected and therefore they take place when the protocol is over, when the honest agents have finished to execute their roles.

2.1.1.8. Coordinated attack problem. We are now able to define what is an attack by several local intruders on a protocol.

Coordinated attack problem (CAP)

Input: A finite set of agents A, list of actions per agent $\{\langle a, l_a\rangle\}_{a\in A}$, a set of local intruders $\mathbb{I} = \{I_i\}_{i=1,\dots,N}$ each with initial knowledge $K^0_{I_i}$, and a partial function ι that assigns at most one intruder $\iota(c)$ to each channel c. and some sensitive data given as a ground term s.

Output: Yes iff there is a sequence of actions performed by agents and local intruders such that, at some point, from the union of intruders knowledge it is possible to derive s.

2.1.2. Reduction of the CAP to the satisfiability of a constraint system

We sketch in this section a non-deterministic reduction of the *coordinated attack problem* to the satisfiability of a constraint system. Briefly put, this reduction guesses an ordering of messages sent and received by the agents. Each intruder diverts all the messages put in a queue he has access to, and all messages received by agents are actually sent by the intruder. A guessed execution may be infeasible if the algorithm guesses an ordering in which an intruder who must send a message is unable to construct a message matching the awaited message pattern. We thus introduce constraints imposing that given a set E of messages known by an intruder, he has to construct a message matching a pattern t.

Definition 2.3. Let $\mathcal{E}$ be an equational theory, E be a set of terms and t be a term. We define the couple (E, t) denoted $E \rhd t$ to be a *constraint* modulo $\mathcal{E}$. A *constraint system modulo* $\mathcal{E}$ is a set

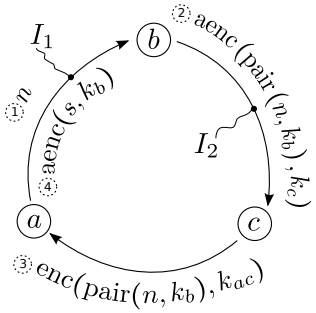
$$\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\dots,n}$$

where n is a non negative integer and $E_i \rhd t_i$ is a constraint modulo $\mathcal{E}$ for $i \in \{1, \dots, n\}$.

A ground substitution σ is a *model* of constraint modulo $\mathcal{E}$ $E \rhd t$ (or σ satisfies this constraint) if $t\sigma \in \mathrm{Der}(E\sigma)$. A ground substitution σ is a *model* of a constraint system $\mathcal{S}$modulo $\mathcal{E}$ if it satisfies all the constraints of $\mathcal{S}$. In order to simplify notations, when the equational theory $\mathcal{E}$ is clear from context we will omit any reference to it.

To express the problem with constraints we first guess an order in which the agents execute their actions since several interleavings are possible. Once the order is fixed, we build the constraint system $\mathcal{S}$ (initially empty) incrementally by progressing in the action list:

- Agent a executes $!_b t$ on channel $a \rightharpoonup b$:
  - if $a \rightharpoonup b$ is controlled by I we increase the knowledge of Intruder I with t by setting $K_I := K_I \cup \{t\}$;
  - otherwise we put t directly in the channel queue.
- Agent a executes $?_b t$ on channel $b \rightharpoonup a$:
  - if $b \rightharpoonup a$ is controlled by I we add a constraint $K_I \rhd t$ into $\mathcal{S}$ expressing the fact that Intruder I knowing K_I must send a message compatible with the expected message pattern t;
  - otherwise we must ensure that the first message m in the queue of $b \rightharpoonup a$ is compatible with the expected pattern t, i.e. $m\sigma = t\sigma$, where σ is a ground substitution satisfying all other constraints in $\mathcal{S}$. For the case of DY deduction system we can encode this



(a) Normal execution + Intruders layout

$$
\begin{aligned}
l_a: &\quad !_b n, \quad ?_c \operatorname{enc}(\operatorname{pair}(n, X), k_{ac}), \quad !_b \operatorname{aenc}(s, X)\\
l_b: &\quad ?_a Y, \quad !_c \operatorname{aenc}(\operatorname{pair}(Y, k_b), k_c), \quad ?_a \operatorname{aenc}(s, k_b)\\
l_c: &\quad ?_b \operatorname{aenc}(Z, k_c), \quad !_a \operatorname{enc}(Z, k_{ac})
\end{aligned}
$$

(b) Sequence of actions per agent

$$K^0_{I_1} \cup \{n\} \rhd Y \tag{1}$$
$$K^0_{I_2} \cup \{\operatorname{aenc}(\operatorname{pair}(Y, k_b), k_c)\} \rhd \operatorname{aenc}(Z, k_c) \tag{2}$$
$$\{\operatorname{enc}(Z, k_{ac})\} \rhd \operatorname{enc}(\operatorname{pair}(n, X), k_{ac}) \tag{3}$$
$$K^0_{I_1} \cup \{n, \operatorname{aenc}(s, X)\} \rhd \operatorname{aenc}(s, k_b) \tag{4}$$
$$K^0_{I_1} \cup \{n, \operatorname{aenc}(s, X)\} \cup K^0_{I_2} \cup \{\operatorname{aenc}(\operatorname{pair}(Y, k_b), k_c)\} \rhd s \tag{5}$$

(c) Constraint system

Figure 2. Example illustrations

equation by a constraint $\{\operatorname{enc}(m, k)\} \rhd \operatorname{enc}(t, k)$ for some atom k [2] since we can prove they admit the same solutions. We add the constraint to $\mathcal{S}$ and remove m from the queue.

Finally, we add $(\bigcup_{I \in \mathbb{I}} K_I) \rhd s$ to $\mathcal{S}$ which expresses the offline communication phase: Secret s can be deduced from the union of local intruders knowledge.

It is clear that if there exists a substitution σ that satisfies $\mathcal{S}$ CAP. Conversely an attack for the CAP orders and instantiates the message patterns in the description of the protocol with a substitution σ, which satisfies the constraint system constructed as above.

We give an example below and will show in Sections 3 and 4 that the satisfiability of constraint systems is in *NP* in the case of the Dolev-Yao deduction theory and its extension with an associative-commutative idempotent operator. This kind of operator can be useful to model messages in XML format as is shown in next subsection.

2.1.3. Example.

Suppose three agents a, b, c execute a protocol whose normal execution is shown in Figure 2(a). Each agent follows his sequence of actions shown in Figure 2(b). Agent a wants to send a secret s to b. He asks b to send her public key k_b to c, that will forward it to a. Nonce n permits a to correlate the message from c with his request.

The agents are connected by three directed channels: channel $a \rightharpoonup b$ is controlled by local intruder I_1, channel $b \rightharpoonup c$ is controlled by local intruder I_2, while channel $c \rightharpoonup a$ is secure (free from intruders).

We assume that the two local intruders have as initial knowledge a pair of fresh public/private keys and public keys of all participants: $K^0_{I_1} = K^0_{I_2} = \{k_i, \operatorname{priv}(k_i), k_a, k_b, k_c\}$.

The question we ask is whether the local intruders I_1 and I_2 can cooperate in such a way that joining their final knowledge they can derive s, although they have no means to communicate during the protocol execution.

Following the sequence of actions shown in Figure 2(a), we obtain the constraint system shown in Figure 2(c). First a sends n which is intercepted by I_1. The knowledge of I_1

[2] for different deduction systems and equational theories the encoding may vary.

becomes $K_{I_1} = K^0_{I_1} \cup \{n\}$. Then I_1 must build from his knowledge a message that can be accepted by b. This is expressed in (1). Once b accepts this message, he executes his next action: emission of a message on channel $b \rightharpoonup c$. This message is intercepted by I_2 and his knowledge becomes $K_{I_2} = K^0_{I_2} \cup \{\text{aenc}(\text{pair}(Y, k_b), k_c)\}$. Now, I_2 must send a message to c on behalf of b (2). Then c accepts a message sent by I_2, it sends his reply to a which should be compatible with the expected pattern: $\text{enc}(Z, k_{ac}) = \text{enc}(\text{pair}(n, X), k_{ac})$. This fact is encoded into (3). Then a sends a message to b intercepted by I_1 ($K_{I_1} = K^0_{I_1} \cup \{n, \text{aenc}(s, X)\}$) and I_1 must generate a message acceptable by b (4). Finally, in (5) we specify that from the common knowledge of I_1 and I_2 one may deduce a secret s.

The obtained constraint system has no solution as well as any other constraint systems generated from different interleaving of actions executed by the agents. We conclude that there is no coordinated attack under the given hypothesis. On the other hand, if I_1 and I_2 can communicate, or the same intruder controls both channels $a \rightharpoonup b$ and $b \rightharpoonup c$, a simple attack can be mounted: once the intruder receives n on channel $a \rightharpoonup b$, he generates the message $\text{aenc}(\text{pair}(n, k_i), k_c)$ and sends it via $b \rightharpoonup c$; c accepts it and sends message $\text{enc}(\text{pair}(n, k_i), k_{ac})$ to a which matches the expected pattern. Thus, a will send via $a \rightharpoonup b$ a message $\text{aenc}(s, k_i)$ which is intercepted by the intruder and can be decomposed using $\text{priv}(k_i)$. The intruder can also finalize the actions of b by sending him, e.g., n and then $\text{aenc}(s, k_b)$.

2.2. Attacks exploiting XML format of messages

We propose a way to model some attacks based on an XML-representation of messages. A different technique to handle this kind of attacks was presented in Chevalier et al. (2007) which considers multisets of XML nodes, while in this work the notion of set is taken as a basis.

We consider an e-shop that accepts e-cheques, and we suppose that it is presented by a Web Service using the SOAP protocol for exchanging messages.

It consists of two services:

(1) the first exposes the list of goods for sale with their prices and processes the orders by accepting payment;
(2) the second is a delivery service; it receives information from the first one about successfully paid orders, and sends the ordered goods to the buyer.

A simple scenario for ordering items is shown in Figure 3. First, a client sends an order using the e-shop interface that consists of an item identifier, e-cheque, delivery address and some comments. Then, the first service of the e-shop checks whether the price of the ordered item corresponds to the received cheque. If it does, the service consumes the cheque and forwards the order to the stock/delivery service (without the used e-cheque). Stock and Delivery service prepares a parcel for the ordered item and sends it to given address. The comment is automatically printed on the parcel to give some information to the postman about, for example, delivery time or access instructions.

Suppose Alice has an e-cheque for 5€. She can buy a simple pen (with ItemID simple) but she likes very much a more expensive gilded one (with ItemID gilded). Can we help Alice to get what she wants for what she has?

Let us formalize the behaviour of scenario players. We will write $(t_1 \cdot \ldots \cdot t_n)$ (or equivalently $\cdot(t_1, \ldots, t_n)$) an XML message where $t_1, \ldots, t_n$ are XML nodes. We model the fact that the nodes are not duplicated and their order is meaningless by assuming that $\cdot$ is

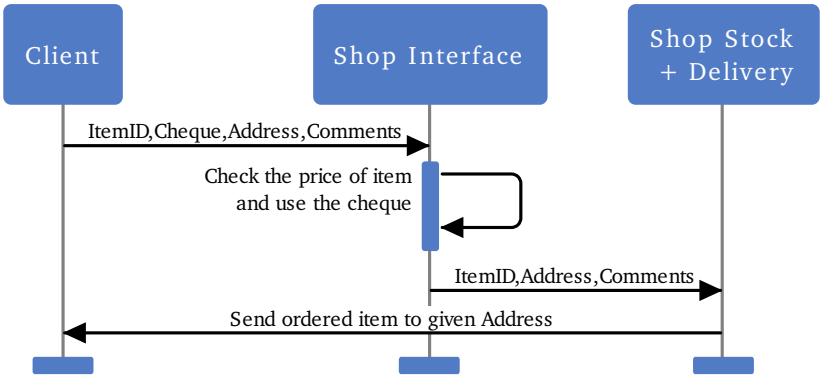


Figure 3. Ordering an item scenario

associative commutative and idempotent. Identifiers starting with a capital letter are considered as variables; numbers and identifier starting from lower-case letter are considered as constants. We model a delivery of item with some *ItemID* to address *Address* with comments *Comments* by the following message: $\text{sig}\left((\mathit{ItemID} \cdot \mathit{Address} \cdot \mathit{Comments}), \text{priv}\left(k_s\right)\right)$ — a message signature produced by e-shop, where k_s its public key and $\text{priv}\left(k_s\right)$ is the corresponding private key, such that no one can produce this message except the shop. We abstract away from the procedure of checking the item price and will suppose that Shop Interface expects a 5€ e-cheque for Item "*simple*". For simplicity we assume only two items.

We will use the same notation as in § 2.1 for emission and reception actions.

For Shop Interface we have:

$$?_{\mathit{Client}}(\mathit{simple} \cdot \mathit{cheque5} \cdot \mathit{IAddr} \cdot \mathit{IComm});$$
$$!_{\mathit{Delivery}}(\mathit{simple} \cdot \mathit{IAddr} \cdot \mathit{IComm}).$$

For Shop Stock/Delivery we have:

$$?_{\mathit{Interface}}(\mathit{DItemID} \cdot \mathit{DAddr} \cdot \mathit{DComm});$$
$$!_{\mathit{Client}}\,\text{sig}\left((\mathit{DItemID} \cdot \mathit{DAddr} \cdot \mathit{DComm}), \text{priv}\left(k_s\right)\right).$$

Alice initially has:

simple, *gilded*:	identifiers of items;
cheque5:	an e-cheque for 5€;
addr:	her address;
cmnts:	residence digital code;
k_s:	a public key of the shop.

Now we build a mixed constraint system (derivation constraints and equations) to

know whether Alice can do what she wants:

$$\left\{\begin{array}{l} \{gilded, simple, cheque5, addr, cmnts, k_s\} \vartriangleright \\ \qquad\qquad (simple \cdot cheque5 \cdot IAddr \cdot IComm) \\ (simple \cdot IAddr \cdot IComm) =_{ACI} (DItemID \cdot DAddr \cdot DComm) \\ \{gilded, simple, cheque5, addr, cmnts, k_s, \\ \text{sig}\,((DItemID \cdot DAddr \cdot DComm), \text{priv}\,(k_s))\} \vartriangleright \\ \qquad\qquad \text{sig}\,((gilded \cdot addr \cdot DComm), \text{priv}\,(k_s)) \end{array}\right. \begin{array}{l} (6) \\ (7) \\ \\ \\ (8) \end{array}$$

The constraint (6) shows that Alice can construct a message expected by the shop from a client. Constraint (7) represents a request from the first to the second service of the shop: the left-hand side is a message sent by the interface service and the right-hand side is a message expected by the stock/delivery subservice; $\equiv_{ACI}$ means that these messages must be compatible (modulo ACI). The last constraint shows that from the received values Alice can build a message that models a delivery of item with ItemID gilded.

To solve it, we first get rid of syntactic equations by applying the most general unifier; and then of equations modulo ACI ($t_1 =_{ACI} t_2$) by encoding them into a deducibility constraint (as it was done in § 2.1.2).

Then, one of the solutions is:

$IAddr$	$\mapsto addr$	$IComm$	$\mapsto (gilded \cdot cmnts)$
$DItemID$	$\mapsto gilded$	$DAddr$	$\mapsto addr$
		$DComm$	$\mapsto (simple \cdot cmnts)$

From this solution we see that Alice can send an ill-formed comment (that contains two XML-nodes), and the Delivery service parser can choose an entry with ID gilded. An attack-request can look like this:

```
<ItemID>simple</ItemID>
<Cheque>cheque5</Cheque>
<Address>addr</Address>
<Comments>cmnts</Comments>
<ItemID>gilded</ItemID>
```

The parser of the first service can return the value of the first occurrence of ItemID: <ItemID>simple</ItemID>, while the parser of the second one, being a different software, can return <ItemID>gilded</ItemID>.

This attack is possible if Alice constructs a request "by hand", but a similar attack is probably feasible using an XML-injection: Alice when filling a request form enters instead of her comments the following string:

```
cmnts</Comments>
<ItemID>gilded</ItemID><Comments>
```

and in the resulting request we get:

```
<ItemID>simple</ItemID>
<Cheque>cheque5</Cheque>
<Address>addr</Address>
<Comments>cmnts</Comments>
```

<ItemID>gilded</ItemID><Comments>
</Comments>

This kind of XML-injection attack was described in OWASP Foundation (2008).

3. Satisfiability of general DY+ACI constraint systems

In Section 2 we have shown how to reduce the problem of protocol insecurity in presence of several intruders to solving a system of general deducibility constraints. In this section we present a decision procedure for the satisfiability of general constraint systems where the Dolev-Yao deduction system is extended with some deduction rules for an associative-commutative-idempotent symbol (DY+ACI). We consider operators for pairing, symmetric and asymmetric encryptions, signature, hashing and an ACI operator that will be used as a set constructor.

As for the proof structure, after introducing the formal notations and some basic properties, the main steps to show the decidability are as follows:

(1) We present an algorithm for solving the derivability problem in the DY+ACI model;
(2) We prove that for checking intruder constraints satisfiability it is sufficient to consider normalized constraints and normalized substitutions;
(3) We show that a satisfiable normalized constraint system admits at least one conservative solution, that is a substitution σ that maps each variable of the constraint system to a set of subterms from the constraint system (instantiated with σ) and private keys;
(4) We give a bound on the size of a conservative solution, and, as a consequence, we obtain decidability.

3.1. Formal introduction to the problem

First we would like to note that we omit the proofs of some statements that we find intuitive. In any case the detailed proofs may be found in Avanesov (2011) (particularly in Lemma 4, page 33). Though the notions of terms and subterms are standard in the litterature, they are not necessarily the most convenient when working with equational theories. Accordingly, in this section and in addition to the standard notions, we introduce the sets of *Subterms* and *Quasi-subterms* of a term. The notion of subterm of a term t is the standard one, whereas quasi-subterms are subterms of the flattening of t by successive applications of associativity rules. The notions of subterms and quasi-subterms are equivalent on normalized terms, but one of the difficulty of handling an equational theory is to analyze their interplay when instantiating and normalizing a term modulo this equational theory. We also provide a few technical lemmas to adress this question in this subsection.

Definition 3.1. Let $\mathcal{A}$ be a set of atoms, and $\mathcal{X}$ be a set of variables. Let $\mathcal{F}$ be the minimal set of functional symbols containing: binary symbols $\mathrm{pair}, \mathrm{enc}, \mathrm{aenc}, \mathrm{sig}, \mathrm{apply}, \cdot_2$, unary symbols $\mathrm{priv}, \cdot_1$, and for all $i \in \mathbb{N}^+$ an i-nary symbol $\cdot_i$. The set of *terms* $\mathcal{T}(\mathcal{A}, \mathcal{X})$ is a minimal set satisfying: $\mathcal{A} \subseteq \mathcal{T}$, $\mathcal{X} \subseteq \mathcal{T}$, if $t_1, \ldots, t_k$ are terms and f is a k-ary symbol from $\mathcal{F}$ then $f(t_1, \ldots, t_k) \in \mathcal{T}$. To be short we write $\mathcal{T}$ instead of $\mathcal{T}(\mathcal{A}, \mathcal{X})$.

By $\mathrm{sig}\,(p, \mathrm{priv}\,(a))$ we mean a signature of message p with private key $\mathrm{priv}\,(a)$ that does not contain a message itself (the message p together with its signature with private key $\mathrm{priv}\,(a)$ can be modeled as $\mathrm{pair}\,(p, \mathrm{sig}\,(p, \mathrm{priv}\,(a)))$).

We consider the following family of function symbols (indexed by their fixed arities): $\bigcup_{i\in\mathbb{N}^+}\{\cdot_i\}$. Symbol $\cdot_i$ has a list of i terms as its arguments. Since the arity can be inferred from the number of arguments, we write $\cdot\,(t_1, t_2)$ as a shortcut for $\cdot_2(t_1, t_2)$, $\cdot\,(t_1, t_2, t_3)$ as a shortcut for $\cdot_3(t_1, t_2, t_3)$ and so on.

We denote $\mathrm{dom}\,(\sigma)$the *domain* $\{x_1, \ldots, x_n\}$ of a substitution $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ and $\mathrm{img}\,(\sigma) = \mathrm{dom}\,(\sigma)\,\sigma = \bigcup_{x\in\mathrm{dom}(\sigma)}\{x\sigma\}$ its *image*. The cardinality of a set P is denoted by $|P|$.

We denote bin any element of $\{\mathrm{enc}, \mathrm{aenc}, \mathrm{pair}, \mathrm{sig}, \mathrm{apply}\}$. The *root* of a term $t \in \mathcal{T}$ is denoted $\mathrm{root}\,(t)$ and is defined as follows:

$$\mathrm{root}\,(t) = \begin{cases} \mathrm{bin}, & \text{if } t = \mathrm{bin}\,(p, q)\,; \\ \cdot, & \text{if } t = \cdot\,(L)\,; \\ \mathrm{priv}, & \text{if } t = \mathrm{priv}\,(p)\,; \\ t, & \text{if } t \in \mathcal{X} \cup \mathcal{A}. \end{cases}$$

We denote the i-th term of a list of terms L as $L[i]$ and we write $t \in L$ if t occurs as an element of L. We also define two binary relations $\subseteq$ and $\approx$ on lists as follows: $L_1 \subseteq L_2$ iff every element of L_1 is an element of L_2; $L_1 \approx L_2$ iff $L_1 \subseteq L_2$ and $L_2 \subseteq L_1$ (we use the same definition if L_1 or L_2 is a set).

In Definition 3.2 we present algebraic properties of the $\cdot$ symbol (denoted as ACI), that are considered together with the Dolev-Yao deduction system in this section. In the rest of this paper deduction constraints are always modulo this ACI equational theory.

Definition 3.2. We define $\equiv_{ACI}$ to be the congruence relation on $\mathcal{T}$ generated by the equational theory ACI:

$$\{\cdot\,(y) = y, \quad \cdot\,(y_1, y_1) = \cdot\,(y_1)\,, \quad \cdot\,(y_1, y_2) = \cdot\,(y_2, y_1)\} \cup \\ \textstyle\bigcup_{n=1}^{\infty}\{\cdot\,(t_1, \ldots, t_k, \cdot\,(t_{k+1}, \ldots, t_m)\,, t_{m+1}, \ldots, t_n) = \cdot\,(t_1, \ldots, t_n)\}_{0\le k<m\le n}\,.$$

Since $\cdot$ is variadic, the associativity property is expressed above as an infinite set of flattening rules.

Definition 3.3. The *set of elements* of a term $t \in \mathcal{T}$ is denoted $\mathrm{elems}\,(t)$ and defined by:

$$\mathrm{elems}\,(t) = \begin{cases} \bigcup_{p\in L}\mathrm{elems}\,(p) & \text{if } t = \cdot\,(L)\,; \\ \{t\}\,, & \text{otherwise.} \end{cases}$$

We extend $\mathrm{elems}\,()$ to sets of terms or lists of terms T by $\mathrm{elems}\,(T) = \bigcup_{t\in T}\mathrm{elems}\,(t)$.

Example 1. The set of elements of $t = \cdot\,(a, \cdot\,(b, a, \mathrm{pair}\,(a, b))\,, \mathrm{pair}\,(\cdot\,(b, b)\,, a))$ is $\mathrm{elems}\,(t) = \{a, b, \mathrm{pair}\,(\cdot\,(b, b)\,, a)\,, \mathrm{pair}\,(a, b)\}$.

Definition 3.4. Let $\prec$ be a strict total order on $\mathcal{T}$ such that the question whether $p \prec q$ can be answered in polynomial time.

Definition 3.5. The *normal form* of a term t (denoted by $\ulcorner t \urcorner$) and a set of terms T (denoted by $\ulcorner T \urcorner$) is defined in a mutually recursive way (on their size) by:

(1) $\ulcorner t \urcorner = t$, if $t \in \mathcal{X} \cup \mathcal{A}$;
(2) $\ulcorner \mathrm{bin}\,(t_1, t_2) \urcorner = \mathrm{bin}\,(\ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner)$;
(3) $\ulcorner \mathrm{priv}\,(t) \urcorner = \mathrm{priv}\,(\ulcorner t \urcorner)$;
(4) $\ulcorner \cdot (L) \urcorner = \begin{cases} \cdot (L'), & \text{if } |\ulcorner \mathrm{elems}\,(L) \urcorner| > 1 \text{ and } L' \approx \ulcorner \mathrm{elems}\,(L) \urcorner \\ & \text{and for all } i < j,\ L'[i] \prec L'[j]; \\ t', & \text{if } \ulcorner \mathrm{elems}\,(L) \urcorner = \{t'\} \end{cases}$
(5) $\ulcorner T \urcorner = \{\ulcorner t \urcorner : t \in T\}$

We say a term t is *normalized* iff $t = \ulcorner t \urcorner$.

We note right away that $|T\sigma| \leq |T|$ and $|\ulcorner T \urcorner| \leq |T|$ and two terms are congruent modulo the ACI properties of $''.''$ iff they have the same normal form. In particular the following equalities hold: $\ulcorner \cdot (t, t) \urcorner = \ulcorner t \urcorner$, $\ulcorner \cdot (t_1, t_2) \urcorner = \ulcorner \cdot (t_2, t_1) \urcorner$, $\ulcorner \cdot (\cdot (t_1, t_2), t_3) \urcorner = \ulcorner \cdot (t_1, \cdot (t_2, t_3)) \urcorner = \ulcorner \cdot (t_1, t_2, t_3) \urcorner$.

Note also that normalization is idempotent: $\ulcorner \ulcorner t \urcorner \urcorner = \ulcorner t \urcorner$. Moreover the equivalence relation $\equiv_{ACI}$ is *closed under substitution* (see Baader and Nipkow (1998)), i.e. $t_1 \equiv_{ACI} t_2$ implies $t_1\sigma \equiv_{ACI} t_2\sigma$. Therefore, we obtain $t \equiv_{ACI} \ulcorner t \urcorner$ and $t\sigma \equiv_{ACI} \ulcorner t \urcorner \sigma$, and then $\ulcorner t\sigma \urcorner = \ulcorner \ulcorner t \urcorner \sigma \urcorner$. Furthermore $\ulcorner t\sigma \urcorner = \ulcorner \ulcorner t \urcorner \sigma \urcorner = \ulcorner t \ulcorner \sigma \urcorner \urcorner = \ulcorner \ulcorner t \urcorner \ulcorner \sigma \urcorner \urcorner$.

Given its importance we turn this last property into a lemma.

Lemma 1. For any term t and substitution σ we have $\ulcorner t\sigma \urcorner = \ulcorner \ulcorner t \urcorner \ulcorner \sigma \urcorner \urcorner$.

Example 2. In Example 1 we have $\ulcorner t \urcorner = \cdot (\{a, b, \mathrm{pair}\,(a, b), \mathrm{pair}\,(b, a)\})$.

We give some useful properties of $\mathrm{elems}\,(\cdot)$ function with respect to normalization.

Lemma 2. For any term t, $\ulcorner \mathrm{elems}\,(t) \urcorner = \mathrm{elems}\,(\ulcorner t \urcorner)$.

Proof. This statement is trivial, if $t \neq \cdot (L)$. Otherwise, let $t = \cdot (t_1, \ldots, t_n)$.
(1) If $\ulcorner \mathrm{elems}\,(t) \urcorner = \{p\}$, where $p \neq \cdot (L_p)$. Then $\ulcorner t \urcorner = p$ and then $\mathrm{elems}\,(\ulcorner t \urcorner) = \mathrm{elems}\,(p) = \{p\} = \ulcorner \mathrm{elems}\,(t) \urcorner$;
(2) If $\ulcorner \mathrm{elems}\,(t) \urcorner = \{p_1, \ldots, p_k\}$, $k > 1$, where $p_i \neq \cdot (L_i)$ for all i. Then $\ulcorner t \urcorner = \cdot (L)$, where $L \approx \{p_1, \ldots, p_k\}$. That means that $\mathrm{elems}\,(\ulcorner t \urcorner) = \bigcup_{p \in \{p_1, \ldots, p_k\}} \mathrm{elems}\,(p) = \{p_1, \ldots, p_k\}$.
□

Lemma 3. For any terms $t_1, \ldots, t_m$, we have $\ulcorner \cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner) \urcorner = \ulcorner \cdot (t_1, \ldots, t_m) \urcorner$

Proof. Follows from the definition of normal form and Lemma 2. □

Lemma 4. For terms $t_1, \ldots, t_m$, we have:
$\mathrm{elems}\,(\ulcorner \cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner) \urcorner) = \mathrm{elems}\,(\cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner)) = \bigcup_{i=1,\ldots,m} \mathrm{elems}\,(\ulcorner t_i \urcorner)$

Proof. We get the first equality by applying Lemma 2: $\mathrm{elems}\,(\ulcorner \cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner) \urcorner) = \ulcorner \mathrm{elems}\,(\cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner)) \urcorner$ and then from Definition 3.3 and Lemma 2 we have that $\mathrm{elems}\,(\cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner))$ is a set of normalized terms. The second equality directly follows from Definition 3.3. □

We introduce now $quasi - subterms$ as subterms of terms flattened by applications of associativity rules.

Definition 3.6. The set of quasi-subterms of term t is defined as follows:

$$\mathrm{QSub}(t) = \begin{cases} \{t\}, & \text{if } t \in \mathcal{X} \cup \mathcal{A}; \\ \{t\} \cup \mathrm{QSub}(t_1), & \text{if } t = \mathrm{priv}(t_1); \\ \{t\} \cup \mathrm{QSub}(t_1) \cup \mathrm{QSub}(t_2), & \text{if } t = \mathrm{bin}(t_1, t_2); \\ \{t\} \cup \bigcup_{p \in \mathrm{elems}(L)} \mathrm{QSub}(p), & \text{if } t = \cdot(L) \end{cases}$$

If T is a set of terms then $\mathrm{QSub}(T) = \bigcup_{t \in T} \mathrm{QSub}(t)$. If $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ is a constraint system then $\mathrm{QSub}(\mathcal{S}) = \bigcup_{t \in \bigcup_{i=1}^{n} E_i \cup \{t_i\}} \mathrm{QSub}(t)$.

Example 3. Referring to Example 1, we have

$$\mathrm{QSub}(t) = \{\cdot(a, \cdot(b, a, \mathrm{pair}(a, b)), \mathrm{pair}(\cdot(b, b), a)), a, b, \mathrm{pair}(a, b), \mathrm{pair}(\cdot(b, b), a), \cdot(b, b)\}.$$

Note also that $\mathrm{QSub}(\mathrm{QSub}(t)) = \mathrm{QSub}(t)$ and $s = \ulcorner s \urcorner$ for all $s \in \mathrm{QSub}(\ulcorner t \urcorner)$.
Let us now define the set $\mathrm{Sub}(t)$ of *subterms* of a term t.

Definition 3.7. Let t be a term. We define $\mathrm{Sub}(t)$ as follows:

$$\mathrm{Sub}(t) = \begin{cases} \{t\}, & \text{if } t \in \mathcal{X} \cup \mathcal{A}; \\ \{t\} \cup \mathrm{Sub}(t_1), & \text{if } t = \mathrm{priv}(t_1); \\ \{t\} \cup \mathrm{Sub}(t_1) \cup \mathrm{Sub}(t_2), & \text{if } t = \mathrm{bin}(t_1, t_2); \\ \{t\} \cup \bigcup_{p \in L} \mathrm{Sub}(p), & \text{if } t = \cdot(L). \end{cases}$$

If T is a set of terms, then $\mathrm{Sub}(T) = \bigcup_{t \in T} \mathrm{Sub}(t)$. If $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ is a constraint system, we define $\mathrm{Sub}(\mathcal{S}) = \bigcup_{t \in \bigcup_{i=1}^{n} E_i \cup \{t_i\}} \mathrm{Sub}(t)$.

Example 4. Referring to Example 1, we have

$$\mathrm{Sub}(t) = \mathrm{QSub}(t) \cup \{\cdot(b, a, \mathrm{pair}(a, b))\}.$$

Lemma 5. For any term t and substitution σ, $\mathrm{Sub}(t\sigma) = \mathrm{Sub}(t)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(t)\,\sigma)$.

Proof. By induction on $|\mathrm{Sub}(t)|$
- $|\mathrm{Sub}(t)| = 1$.
  - $t \in \mathcal{A}$. As $t\sigma = t$ and $\mathrm{Vars}(t) = \emptyset$, the statement becomes trivial;
  - $t \in \mathcal{X}$. Then $\mathrm{Sub}(t)\,\sigma = t\sigma$, $\mathrm{Vars}(t) = \{t\}$; and as for any term p, $p \in \mathrm{Sub}(p)$, we have $\mathrm{Sub}(t\sigma) = \{t\sigma\} \cup \mathrm{Sub}(t\sigma)$.
- Suppose that for any $t : |\mathrm{Sub}(t)| < k$ $(k \geq 1)$, the statement is true;
- Given a term $t : |\mathrm{Sub}(t)| = k$, $k > 1$. Let us consider all possible cases:
  - $t = \mathrm{bin}(t_1, t_2)$.Then $t\sigma = \mathrm{bin}(t_1\sigma, t_2\sigma)$ and $\mathrm{Vars}(t) = \mathrm{Vars}(t_1) \cup \mathrm{Vars}(t_2)$. $\mathrm{Sub}(t\sigma) = \{t\sigma\} \cup \mathrm{Sub}(t_1\sigma) \cup \mathrm{Sub}(t_2\sigma) = (\text{as } |\mathrm{Sub}(t_i)| < k) = \{t\sigma\} \cup \mathrm{Sub}(t_1)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(t_1)\,\sigma) \cup \mathrm{Sub}(t_2)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(t_2)\,\sigma) = \{t\sigma\} \cup \mathrm{Sub}(t_1)\,\sigma \cup \mathrm{Sub}(t_2)\,\sigma \cup \mathrm{Sub}((\mathrm{Vars}(t_1) \cup \mathrm{Vars}(t_2))\sigma) = \mathrm{Sub}(t)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(t)\,\sigma)$;
  - $t = \mathrm{priv}(t_1)$. The proof is the same as for the previous case;
  - $t = \cdot(t_1, \ldots, t_m)$. Then $t\sigma = \cdot(t_1\sigma, \ldots, t_m\sigma)$ and $\mathrm{Vars}(t) = \bigcup_{i=1,\ldots,m} \mathrm{Vars}(t_i)$. Then we have $\mathrm{Sub}(t\sigma) = \{t\sigma\} \cup \bigcup_{i=1,\ldots,m} \mathrm{Sub}(t_i\sigma) = (\text{as } |\mathrm{Sub}(t_i)| < k) = \{t\sigma\} \cup \bigcup_{i=1,\ldots,m} (\mathrm{Sub}(t_i)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(t_i)\,\sigma)) = \{t\sigma\} \cup \bigcup_{i=1,\ldots,m} \mathrm{Sub}(t_i)\,\sigma \cup \cup\, \mathrm{Sub}\left(\left(\bigcup_{i=1,\ldots,m} \mathrm{Vars}(t_i)\right)\sigma\right) = \mathrm{Sub}(t)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(t)\,\sigma)$.

□

The inclusion relation between $\mathrm{elems}(\cdot)$, $\mathrm{QSub}(\cdot)$ and $\mathrm{Sub}(\cdot)$ is stated in Lemma 6.

Lemma 6. Let t be a term. Then $\mathrm{elems}(t) \subseteq \mathrm{QSub}(t) \subseteq \mathrm{Sub}(t)$

For normalized terms, the set of its subterms coincides with the set of its quasi-subterms:

Lemma 7. For any normalized term t, $\mathrm{QSub}(t) = \mathrm{Sub}(t)$.

Proof. By induction on $|\mathrm{Sub}(t)|$.
- $|\mathrm{Sub}(t)| = 1$. Then $t \in \mathcal{X} \cup \mathcal{A}$, and thus, $\mathrm{QSub}(t) = \mathrm{Sub}(t) = \{t\}$;
- Suppose that for any $t : |\mathrm{Sub}(t)| < k$ $(k > 1)$, $\mathrm{QSub}(t) = \mathrm{Sub}(t)$;
- Given a term $t : |\mathrm{Sub}(t)| = k$, $k > 1$. We need to show that $\mathrm{QSub}(t) = \mathrm{Sub}(t)$.
  - $t = \mathrm{bin}(t_1, t_2)$. Then $\mathrm{QSub}(\mathrm{bin}(t_1, t_2)) = \{t\} \cup \mathrm{QSub}(t_1) \cup \mathrm{QSub}(t_2) = (\text{as } |\mathrm{Sub}(t_i)| < k) = \{t\} \cup \mathrm{Sub}(t_1) \cup \mathrm{Sub}(t_2) = \mathrm{Sub}(t)$;
  - $t = \mathrm{priv}(t_1)$. Then $\mathrm{QSub}(\mathrm{priv}(t_1)) = \{t\} \cup \mathrm{QSub}(t_1) = \{t\} \cup \mathrm{Sub}(t_1) = \mathrm{Sub}(t)$;
  - $t = \cdot(L)$. As t is normalized, $\forall p \in L$, $\mathrm{root}(p) \neq \cdot$. Then $\mathrm{elems}(L) \approx L$. Thus, we have $\mathrm{QSub}(t) = \{t\} \cup \bigcup_{p \in \mathrm{elems}(L)} \mathrm{QSub}(p) = \{t\} \cup \bigcup_{p \in L} \mathrm{QSub}(p) = \{t\} \cup \bigcup_{p \in L} \mathrm{Sub}(p) = \mathrm{Sub}(t)$.

□

Lemma 8. For any term t, $\mathrm{QSub}(\ulcorner t \urcorner) = \ulcorner \mathrm{QSub}(t) \urcorner$.

Proof. By induction on $|\mathrm{Sub}(t)|$.
- $|\mathrm{Sub}(t)| = 1$. Then $t \in \mathcal{A} \cup \mathcal{X}$. As $\mathrm{QSub}(t) = \{t\}$ and $t = \ulcorner t \urcorner$, the statement holds;
- Suppose that for any $t : |\mathrm{Sub}(t)| < k$ $(k > 1)$, the statement is true;
- Given t such that $|\mathrm{Sub}(t)| = k$, $k \geq 1$, let us consider all possible cases:

(1) $t = \mathrm{bin}(t_1, t_2)$.On the one hand, $\mathrm{QSub}(t) = \{t\} \cup \mathrm{QSub}(t_1) \cup \mathrm{QSub}(t_2)$. On the other hand, $\ulcorner t \urcorner = \mathrm{bin}(\ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner)$ and then, $\mathrm{QSub}(\ulcorner t \urcorner) = \{\ulcorner t \urcorner\} \cup \mathrm{QSub}(\ulcorner t_1 \urcorner) \cup \mathrm{QSub}(\ulcorner t_2 \urcorner)$. Then, as $\mathrm{QSub}(\ulcorner t_i \urcorner) = \ulcorner \mathrm{QSub}(t_i) \urcorner$, we have that $\mathrm{QSub}(\ulcorner t \urcorner) = \ulcorner \mathrm{QSub}(t) \urcorner$;

(2) $t = \mathrm{priv}(t_1)$. The proof is the same as for previous case;

(3) $t = \cdot(L)$. We have $\mathrm{QSub}(t) = \{t\} \cup \bigcup_{p \in \mathrm{elems}(L)} \mathrm{QSub}(p)$; From Lemma 2 we have $\mathrm{elems}(\ulcorner \cdot(L) \urcorner) = \ulcorner \mathrm{elems}(\cdot(L)) \urcorner$, and then we obtain $\mathrm{QSub}(\ulcorner \cdot(L) \urcorner) = \{\ulcorner \cdot(L) \urcorner\} \cup \bigcup_{p \in \mathrm{elems}(\ulcorner \cdot(L) \urcorner)} \mathrm{QSub}(p) = \ulcorner \{\cdot(L)\} \urcorner \cup \bigcup_{p \in \mathrm{elems}(\cdot(L))} \mathrm{QSub}(\ulcorner p \urcorner) = (\text{by supposition}) = \ulcorner \{\cdot(L)\} \urcorner \cup \bigcup_{p \in \mathrm{elems}(\cdot(L))} \ulcorner \mathrm{QSub}(p) \urcorner = \ulcorner \{\cdot(L)\} \cup \bigcup_{p \in \mathrm{elems}(\cdot(L))} \mathrm{QSub}(p) \urcorner = \ulcorner \mathrm{QSub}(t) \urcorner$.

□

Lemma 9. For any term t, $\mathrm{Sub}(\ulcorner t \urcorner) \subseteq \ulcorner \mathrm{Sub}(t) \urcorner$.

Proof. From Lemma 7 we have $\mathrm{Sub}(\ulcorner t \urcorner) = \mathrm{QSub}(\ulcorner t \urcorner)$. By Lemma 8 we obtain $\mathrm{QSub}(\ulcorner t \urcorner) = \ulcorner \mathrm{QSub}(t) \urcorner$ and by Lemma 6 we have $\mathrm{QSub}(t) \subseteq \mathrm{Sub}(t)$. Thus, $\mathrm{Sub}(\ulcorner t \urcorner) \subseteq \ulcorner \mathrm{QSub}(t) \urcorner \subseteq \ulcorner \mathrm{Sub}(t) \urcorner$. □

Table 2. DY+ACI deduction rules

Composition rules	Decomposition rules
$t_1, t_2 \to \ulcorner \mathrm{enc}\,(t_1, t_2) \urcorner$	$\mathrm{enc}\,(t_1, t_2)\,, t_3 \to \ulcorner t_1 \urcorner$, if $t_2 \equiv_{ACI} t_3$
$t_1, t_2 \to \ulcorner \mathrm{aenc}\,(t_1, t_2) \urcorner$	$\mathrm{aenc}\,(t_1, t_2)\,, \mathrm{priv}\,(t_3) \to \ulcorner t_1 \urcorner$, if $t_2 \equiv_{ACI} t_3$
$t_1, t_2 \to \ulcorner \mathrm{pair}\,(t_1, t_2) \urcorner$	$\mathrm{pair}\,(t_1, t_2) \to \ulcorner t_1 \urcorner$
$t_1, \mathrm{priv}\,(t_2) \to \ulcorner \mathrm{sig}\,(t_1, \mathrm{priv}\,(t_2)) \urcorner$	$\mathrm{pair}\,(t_1, t_2) \to \ulcorner t_2 \urcorner$
$t_1, \ldots, t_m \to \ulcorner \cdot\,(t_1, \ldots, t_m) \urcorner$	$\cdot\,(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$ for all i
$t_1, t_2 \to \ulcorner \mathrm{apply}\,(t_1, t_2) \urcorner$	

3.2. Dolev-Yao deduction system modulo ACI

We define a Dolev-Yao deduction system modulo ACI (denoted DY+ACI). It consists of composition rules and decomposition rules, depicted in Table 2 where $t_1, t_2, \ldots, t_m \in \mathcal{T}$ (i.e., we consider an infinite system having a finite representation given in the table). Note that rule $\mathrm{enc}\,(t_1, t_2)\,, t_3 \to \ulcorner t_1 \urcorner$, if $t_2 \equiv_{ACI} t_3$ is equivalent to $\mathrm{enc}\,(t_1, t_2)\,, \ulcorner t_2 \urcorner \to \ulcorner t_1 \urcorner$, since it is always possible to normalize term t_3 (and obtain $\ulcorner t_3 \urcorner = \ulcorner t_2 \urcorner$) using, e.g. $t_3, t_3 \to \ulcorner \mathrm{pair}\,(t_3, t_3) \urcorner$ and since $\ulcorner \mathrm{pair}\,(t_3, t_3) \urcorner = \mathrm{pair}\,(\ulcorner t_3 \urcorner, \ulcorner t_3 \urcorner)$ we can decompose it by obtaining $\ulcorner \ulcorner t_3 \urcorner \urcorner = \ulcorner t_3 \urcorner$.

We introduce two restrictions to define a subset of *valid* ground terms. The first one follows from the semantics of the functional symbol sig: its second argument must always be a private key. The second one is imposed in order to simplify our reasoning for solving constraint systems: we only consider atomic asymmetric private keys (this is not a real limitation for many applications).

Definition 3.8. A ground term t is *valid* iff for any subterm $s \in \mathrm{Sub}\,(t)$:
(1) $s = \mathrm{sig}\,(p, q) \implies \ulcorner q \urcorner = \mathrm{priv}\,(q')$ for some q' ;
(2) $s = \mathrm{priv}\,(p) \implies \ulcorner p \urcorner \in \mathcal{A}$

We suppose hereinafter that the considered constraint system $\mathcal{S}$ contains at least one atom, i.e. $\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A} \neq \emptyset$. Otherwise we add a dummy constraint $\{a\} \rhd a$ to $\mathcal{S}$ which will be satisfied by any substitution. We define $\mathrm{priv}\,(T) = \{\mathrm{priv}\,(t) : t \in T\}$ for a set of terms T. We define $\mathrm{Vars}\,(\mathcal{S}) = \bigcup_{i=1}^{n} \mathrm{Vars}\,(E_i) \cup \mathrm{Vars}\,(t_i)$. We say that $\mathcal{S}$ is normalized, iff for all $t \in \mathrm{Sub}\,(\mathcal{S})$, t is normalized.

Example 5. We give a example of general constraint system and its solution within DY+ACI deduction system.

$$\mathcal{S} = \left\{ \begin{array}{ll} \mathrm{enc}\,(x, a)\,, \mathrm{pair}\,(c, a) & \rhd\ b \\ \cdot\,(x, c) & \rhd\ a \end{array} \right\},$$

where $a, b, c \in \mathcal{A}$ and $x \in \mathcal{X}$. One of its solutions within DY+ACI is $\sigma = \{x \mapsto \mathrm{enc}\,(\mathrm{pair}\,(a, b)\,, c)\}$.

Definition 3.9. Let $T = \{t_1, \ldots, t_k\}$ be a non-empty set of terms. Then we define $\pi(T)$ as follows:

$$\pi(T) = \ulcorner \cdot\,(t_1, \ldots, t_k) \urcorner .$$

We remark that $\pi(\{t\}) = \ulcorner t \urcorner$ and $\pi(T) = \ulcorner \pi(T) \urcorner$. Note also that $\pi(t_1, \ldots, t_k)$ can be interpreted as a set of terms. The terms derivable from $\pi(t_1, \ldots, t_k)$ are derivable from $t_1, \ldots, t_k$ and vice-versa.

Lemma 10. Let T be a set of terms $T = \{t_1, \ldots, t_k\}$. Then $\pi(T) \in \mathrm{Der}(\ulcorner T \urcorner)$ and $\ulcorner T \urcorner \subseteq \mathrm{Der}(\{\pi(T)\})$.

From Lemma 3 we derive another property: $\pi(T) = \pi(\ulcorner T \urcorner)$.

Lemma 11. $\pi(T_1 \cup \cdots \cup T_m) = \pi(\{\pi(T_1), \ldots, \pi(T_m)\})$

Proof. From definition of π and Lemma 2, we have $\mathrm{elems}(\pi(T_i)) = \ulcorner \mathrm{elems}(T_i) \urcorner$. Next $\pi(\{\pi(T_1), \ldots, \pi(T_m)\}) = \ulcorner \cdot(L) \urcorner$ where

$$\begin{aligned} L &\approx \ulcorner \mathrm{elems}(\{\pi(T_1), \ldots, \pi(T_m)\}) \urcorner \\ &= \ulcorner \bigcup_{i=1,\ldots,m} \ulcorner \mathrm{elems}(T_i) \urcorner \urcorner \\ &= \ulcorner \bigcup_{i=1,\ldots,m} \mathrm{elems}(T_i) \urcorner \end{aligned}$$

On the other hand $\pi(T_1 \cup \cdots \cup T_m) = \ulcorner \cdot(L') \urcorner$, where $L' \approx \ulcorner \bigcup_{i=1,\ldots,m} \mathrm{elems}(T_i) \urcorner$. □

3.3. Solving the derivability problem of DY+ACI

The decidability of the derivability problem for DY has been extensively studied (see *e.g.* Amadio et al. (2000) for the atomic keys case, and Rusinowitch and Turuani (2003) for the complex keys case) and is known to be polynomial. Also, we note that the derivability problem when the set of symbols only contains the dot $\cdot$ and constants is trivial, for in this case a constraint $E \rhd t$ is satisfied if, and only if, the constants occurring in t also occur in E (proof left to the reader). Then the combination algorithm of Chevalier and Rusinowitch (2005) yields directly the decidability of derivability for the DY+ACI case. Moreover since the derivability in DY and ACI sub-theories can be decided in polynomial time the algorithm deciding derivability returns in time polynomial in the number of subterms Cortier and Delaune (2012).

3.4. Existence of conservative solutions for satisfiable systems

In this subsection we show that for any satisfiable constraint system there exist particular models called *conservative solutions*. Roughly speaking, such a model can be defined by mapping each variable to a set of subterms and atoms extracted from the constraint system. This will bound the search space for finding a model (see § 3.5).

First, in Proposition 1 we state that a constraint system and its normal form have the same models and we show the equivalence between the existence of a model and the existence of a normalized model. As a consequence we will need only to consider normalized constraints and models in the sequel.

Proposition 1. *The substitution σ is a model of the constraint system $\mathcal{S}$ if and only if σ is a model of $\ulcorner \mathcal{S} \urcorner$. Moreover, σ is a model of $\mathcal{S}$ if and only if $\ulcorner \sigma \urcorner$ is a model of $\mathcal{S}$.*

Proof. By definition, σ is a model of $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\dots,n}$, iff $\forall i \in \{1,\dots,n\}, \ulcorner t_i\sigma \urcorner \in \mathrm{Der}(\ulcorner E_i\sigma \urcorner)$. But we know that $\ulcorner t_i\sigma \urcorner = \ulcorner \ulcorner t_i \urcorner \sigma \urcorner$ and $\ulcorner E_i\sigma \urcorner = \ulcorner \ulcorner E_i \urcorner \sigma \urcorner$. Thus, σ is a model of $\mathcal{S}$ if and only if σ is a model of $\ulcorner \mathcal{S} \urcorner$. But again, since $\ulcorner t_i\sigma \urcorner = \ulcorner t_i \ulcorner \sigma \urcorner \urcorner$ and $\ulcorner E_i\sigma \urcorner = \ulcorner E_i \ulcorner \sigma \urcorner \urcorner$, we have σ is a model of $\mathcal{S}$ if and only if $\ulcorner \sigma \urcorner$ is a model of $\mathcal{S}$. □

We introduce a transformation $\pi(H^{\mathcal{S},\sigma}(\cdot))$ on ground terms that replaces recursively every occurrence of any binary symbol bin by the ACI symbol $\cdot$ (and flattens nested ACI terms) when this occurrence roots a subterm that is not matched by any non-variable subterm of the constraint system with substitution σ.

Later, we will show that $\pi(\mathrm{H}(\sigma))$ is also a model of $\mathcal{S}$.

Definition 3.10. We denote a set of non-variable subterms of a constraint system $\mathcal{S}$ as $\mathring{\mathrm{Sub}}(S)$, i.e. $\mathrm{Sub}(\mathcal{S}) \setminus \mathcal{X} = \mathring{\mathrm{Sub}}(S)$.

Definition 3.11. Let $\mathcal{S}$ be a normalized constraint system which is satisfiable with the model σ. Let us fix some $\alpha \in (\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))$. We define a function $H^{\mathcal{S},\sigma}(\cdot) : \mathcal{T}_g \to 2^{\mathcal{T}_g}$ as follows:

$$H^{\mathcal{S},\sigma}(t) = \begin{cases} \{\alpha\}, \text{ if } t \in (\mathcal{A} \setminus \mathrm{Sub}(\mathcal{S})); \\ \{a\}, \text{ if } t = a \in (\mathcal{A} \cap \mathrm{Sub}(\mathcal{S})); \\ \{\mathrm{priv}(\pi(H^{\mathcal{S},\sigma}(t_1)))\}, \text{ if } t = \mathrm{priv}(t_1); \\ \{\mathrm{bin}(\pi(H^{\mathcal{S},\sigma}(t_1)), \pi(H^{\mathcal{S},\sigma}(t_2)))\}, \text{ if } t = \mathrm{bin}(t_1,t_2) \\ \qquad \text{and } \ulcorner t \urcorner \in \ulcorner \mathring{\mathrm{Sub}}(S)\sigma \urcorner; \\ H^{\mathcal{S},\sigma}(t_1) \cup H^{\mathcal{S},\sigma}(t_2), \text{ if } t = \mathrm{bin}(t_1,t_2) \\ \qquad \text{and } \ulcorner t \urcorner \notin \ulcorner \mathring{\mathrm{Sub}}(S)\sigma \urcorner; \\ \bigcup_{p \in L} H^{\mathcal{S},\sigma}(p), \text{ if } t = \cdot(L). \end{cases}$$

Henceforward, we will omit parameters and write $H(\cdot)$ instead of $H^{\mathcal{S},\sigma}(\cdot)$ for shorter notation.

Definition 3.12. We define the superposition of $\pi(\cdot)$ and $H(\cdot)$ on a set of terms $T = \{t_1,\dots,t_k\}$ as follows: $\pi(H(T)) = \{\pi(H(t)) \mid t \in T\}$.

Definition 3.13. Let $\theta = \{x_1 \mapsto t_1, \dots, x_k \mapsto t_k\}$ be a substitution. We define $\pi(H(\theta))$ to be the substitution $\{x_1 \mapsto \pi(H(t_1)), \dots, x_k \mapsto \pi(H(t_k))\}$.

Note that since for every $x \in \mathrm{dom}(\theta)$ we have $x \notin \mathrm{Vars}(x\theta)$, we also have $\mathrm{dom}(\pi(H(\theta))) = \mathrm{dom}(\theta)$. Moreover, by definition, $\pi(H(\theta))$ is normalized.

Example 6. Let us consider a (normalized) constraint system $\mathcal{S}$ and its model σ from Example 5 and show that $\pi(H(\sigma))$ is also a model of $\mathcal{S}$. $\pi(H(\mathrm{enc}(\mathrm{pair}(a,b),c))) = \pi(H(\mathrm{pair}(a,b)) \cup \{c\}) = \pi(\{a\} \cup \{b\} \cup \{c\}) = \cdot(a,b,c)$ (we suppose that $a \prec b \prec c$). One can see that $\pi(\mathrm{H}(\sigma)) = \{x \mapsto \cdot(a,b,c)\}$ is also a model of $\mathcal{S}$ within DY+ACI.

We give several useful properties about the function $H(\cdot)$.

Lemma 12. For a ground term t, $H(t) = \bigcup_{p \in \mathrm{elems}(t)} H(p)$.

Lemma 13. For a ground term t, $H(t) = H(\ulcorner t \urcorner)$.

Proof. By induction on $|\mathrm{Sub}(t)|$:
- $|\mathrm{Sub}(t)| = 1$ is possible in the only case: $t = a \in \mathcal{A}$ and as $a = \ulcorner a \urcorner$, the equality is trivial;
- Suppose that for any $t : |\mathrm{Sub}(t)| < k$ $(k > 1)$, $H(t) = H(\ulcorner t \urcorner)$ holds;
- Given a term $t : |\mathrm{Sub}(t)| = k$, $k > 1$. We need to prove that $H(t) = H(\ulcorner t \urcorner)$.
  - if $t = \mathrm{priv}(t_1)$, then $H(t) = \{\mathrm{priv}(\pi(H(t_1)))\} =$ (by induction hypothesis) $= \{\mathrm{priv}(\pi(H(\ulcorner t_1 \urcorner)))\} = H(\mathrm{priv}(\ulcorner t_1 \urcorner)) = H(\ulcorner t \urcorner)$;
  - if $t = \mathrm{bin}(p, q)$ and $\ulcorner t \urcorner \in \ulcorner \mathring{\mathrm{Sub}}(\mathcal{S})\, \sigma \urcorner$; Then $H(\ulcorner t \urcorner) = H(\mathrm{bin}(\ulcorner p \urcorner, \ulcorner q \urcorner)) = \{\mathrm{bin}(\pi(H(\ulcorner p \urcorner)), \pi(H(\ulcorner q \urcorner)))\} =$ (by induction) $= \{\mathrm{bin}(\pi(\ulcorner H(p) \urcorner), \pi(\ulcorner H(q) \urcorner))\} =$ (by Lemma 3) $= \{\mathrm{bin}(\pi(H(p)), \pi(H(q)))\} = H(\mathrm{bin}(p, q))$;
  - if $t = \mathrm{bin}(p, q)$ and $\ulcorner t \urcorner \notin \ulcorner \mathring{\mathrm{Sub}}(\mathcal{S})\, \sigma \urcorner$; Then $H(t) = H(p) \cup H(q) =$ (by induction hypothesis) $= H(\ulcorner p \urcorner) \cup H(\ulcorner q \urcorner) =$ (as $\ulcorner \mathrm{bin}(\ulcorner p \urcorner, \ulcorner q \urcorner) \urcorner = \ulcorner t \urcorner \notin \ulcorner \mathring{\mathrm{Sub}}(\mathcal{S})\, \sigma \urcorner$) $= H(\mathrm{bin}(\ulcorner p \urcorner, \ulcorner q \urcorner)) = H(\ulcorner t \urcorner)$;
  - if $t = \cdot(L)$, where $L = t_1, \ldots, t_m$. Note first that as $t = \cdot(L)$, $\forall s \in \mathrm{elems}(t)$, $|\mathrm{Sub}(s)| < |\mathrm{Sub}(t)|$. Then, by Lemma 12, $H(t) = \bigcup_{p \in \mathrm{elems}(t)} H(p) =$ (by induction hypothesis) $= \bigcup_{p \in \mathrm{elems}(t)} H(\ulcorner p \urcorner)$. On the other part we have $H(\ulcorner t \urcorner) = \bigcup_{p \in \mathrm{elems}(\ulcorner t \urcorner)} H(p) =$ (by Lemma 2) $= \bigcup_{p \in \ulcorner \mathrm{elems}(t) \urcorner} H(p) = \bigcup_{p \in \mathrm{elems}(t)} H(\ulcorner p \urcorner) = H(t)$.

□

As a consequence, $\pi(H(t)) = \pi(H(\ulcorner t \urcorner)) = \ulcorner \pi(H(t)) \urcorner = \ulcorner \pi(H(\ulcorner t \urcorner)) \urcorner$.

We also note as another consequence that $\pi(H(\cdot))$ transforms valid terms to valid ones.

Now, we show that for the subterms of a normalized constraint system instantiated with its normalized model, the application of the transformation $\pi(H(\cdot))$ on such terms is equivalent modulo ACI to the result obtained if the transformation was applied only on the model.

Proposition 2. *Let $\mathcal{S}$ be a normalized constraint system and σ its normalized model. For all $t \in \mathrm{Sub}(\mathcal{S})$, $\ulcorner t\, \pi(\mathrm{H}(\sigma)) \urcorner = \pi(H(t\sigma))$.*

Proof. Note that t is normalized. We will prove it by induction on $|\mathrm{Sub}(t)|$.
- Let $|\mathrm{Sub}(t)| = 1$. Then:
  - either $t \in \mathcal{A}$. In this case $t \in (\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))$, and as $t\mu = t$ for any substitution μ, then $\pi(H(t\sigma)) = \pi(H(t)) = \pi(\{t\}) = t$ and $t\, \pi(\mathrm{H}(\sigma)) = t$. Thus, $t\, \pi(\mathrm{H}(\sigma)) = \pi(H(t\sigma))$.
  - or $t \in \mathcal{X}$. As σ is a model and $t \in \mathrm{Sub}(\mathcal{S})$, we have $t \in \mathrm{dom}(\sigma)$, and, by definition, $t \in \mathrm{dom}(\pi(\mathrm{H}(\sigma)))$. Then, by definition of $\pi(\mathrm{H}(\sigma))$, $t\, \pi(\mathrm{H}(\sigma)) = \pi(H(t\sigma))$.
- Assume that for some $k \geq 1$ if $|\mathrm{Sub}(t)| \leq k$, then $\ulcorner t\, \pi(\mathrm{H}(\sigma)) \urcorner = \ulcorner \pi(H(t\sigma)) \urcorner$.
- Show that for any t such that $|\mathrm{Sub}(t)| \geq k + 1$, where $t = \mathrm{bin}(p, q)$ or $t = \mathrm{priv}(q)$ or $t = \cdot(t_1, \ldots, t_m)$, but $|\mathrm{Sub}(p)| \leq k$, $|\mathrm{Sub}(q)| \leq k$ and $|\mathrm{Sub}(t_i)| \leq k$, for all $i \in \{1, \ldots, m\}$, statement $\ulcorner t\, \pi(\mathrm{H}(\sigma)) \urcorner = \ulcorner \pi(H(t\sigma)) \urcorner$ is still true. We have:
  - either $t = \mathrm{bin}(p, q)$. As $t = \mathrm{bin}(p, q) \in \mathrm{Sub}(\mathcal{S}) \Rightarrow p \in \mathrm{Sub}(\mathcal{S})$ and $q \in \mathrm{Sub}(\mathcal{S})$. As $|\mathrm{Sub}(p)| < |\mathrm{Sub}(t)|$ and from the induction assumption, we have $\ulcorner p\, \pi(\mathrm{H}(\sigma)) \urcorner = \ulcorner \pi(H(p\sigma)) \urcorner$. The same holds for q.

Again, as $\mathrm{bin}(p,q)\,\sigma \in \mathring{\mathrm{Sub}}(S)\,\sigma$ (as $\mathrm{bin}(p,q) \notin \mathcal{X}$ and $t \in \mathrm{Sub}(\mathcal{S})$) we have that $\ulcorner\pi(H(\mathrm{bin}(p,q)\,\sigma))\urcorner = \ulcorner\pi(H(\mathrm{bin}(p\sigma,q\sigma)))\urcorner = \ulcorner\pi(H(\ulcorner\mathrm{bin}(p\sigma,q\sigma)\urcorner))\urcorner = \ulcorner\pi(H(\mathrm{bin}(\ulcorner p\sigma\urcorner,\ulcorner q\sigma\urcorner)))\urcorner = \ulcorner\pi(\{\mathrm{bin}(\pi(H(\ulcorner p\sigma\urcorner)),\pi(H(\ulcorner q\sigma\urcorner)))\})\urcorner = \ulcorner\pi(\{\mathrm{bin}(\ulcorner\pi(H(p\sigma))\urcorner,\ulcorner\pi(H(q\sigma))\urcorner)\})\urcorner = \ulcorner\mathrm{bin}(\ulcorner p\,\pi(\mathrm{H}(\sigma))\urcorner,\ulcorner q\,\pi(\mathrm{H}(\sigma))\urcorner)\urcorner = \ulcorner\mathrm{bin}(p\,\pi(\mathrm{H}(\sigma)),q\,\pi(\mathrm{H}(\sigma)))\urcorner = \ulcorner\mathrm{bin}(p,q)\,\pi(\mathrm{H}(\sigma))\urcorner = \ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner$.

- or $t = \cdot(t_1,\ldots,t_m)$. We have for all $i \in \{1,\ldots,m\}$, $t_i \in \mathrm{Sub}(\mathcal{S})$, thus, by induction hypothesis $\pi(H(t_i\sigma)) = \ulcorner t_i\,\pi(\mathrm{H}(\sigma))\urcorner$. $\pi(H(t\sigma)) = \pi(H(\cdot(t_1\sigma,\ldots,t_m\sigma))) = \pi(H(t_1\sigma)\cup\cdots\cup H(t_m\sigma)) =$ (by Lemma 11) $= \pi(\{\pi(H(t_1\sigma)),\ldots,\pi(H(t_m\sigma))\}) = \pi(\{\ulcorner t_1\,\pi(\mathrm{H}(\sigma))\urcorner,\ldots,\ulcorner t_m\,\pi(\mathrm{H}(\sigma))\urcorner\}) = \ulcorner\cdot(\ulcorner t_1\,\pi(\mathrm{H}(\sigma))\urcorner,\ldots,\ulcorner t_m\,\pi(\mathrm{H}(\sigma))\urcorner)\urcorner = \ulcorner\cdot(t_1\,\pi(\mathrm{H}(\sigma)),\ldots,t_m\,\pi(\mathrm{H}(\sigma)))\urcorner = \ulcorner(\cdot(t_1,\ldots,t_m))\,\pi(\mathrm{H}(\sigma))\urcorner = \ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner$
- or $t = \mathrm{priv}(q)$. Then $q \in \mathrm{Sub}(\mathcal{S})$.
  $\pi(H(t\sigma)) = \pi(\{\mathrm{priv}(\pi(H(q\sigma)))\}) = \ulcorner\mathrm{priv}(\pi(H(q\sigma)))\urcorner = \ulcorner\mathrm{priv}(q\,\pi(\mathrm{H}(\sigma)))\urcorner = \ulcorner\mathrm{priv}(q)\,\pi(\mathrm{H}(\sigma))\urcorner = \ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner$.

Thus, the proposition is proven. □

Now we show that the derivability of a term from a set of terms is preserved by the transformation $\pi(H(\cdot))$. But before we recall two simple properties of the derivability relation:

Lemma 14. Let $A,B,C,D \subseteq \mathcal{T}_g$. Then if $A \subseteq \mathrm{Der}(B)$ and $B \subseteq \mathrm{Der}(C)$ then $A \subseteq \mathrm{Der}(C)$. Moreover, if $A \subseteq \mathrm{Der}(B)$ and $C \subseteq \mathrm{Der}(D)$ then $A \cup C \subseteq \mathrm{Der}(B \cup D)$.

Proposition 3. *Let $\mathcal{S}$ be a normalized constraint system and σ its normalized model. For any DY+ACI rule $l_1,\ldots,l_k \to r$, $\pi(H(r)) \in \mathrm{Der}(\{\pi(H(l_1)),\ldots,\pi(H(l_k))\})$.*

Proof idea. We proceed by considering all possible deduction rules. To give an idea, we show a proof for only one rule (see the full proof in Appendix B.4): $\mathrm{aenc}(t_1,t_2), \ulcorner\mathrm{priv}(t_2)\urcorner \to \ulcorner t_1\urcorner$. We have to show that term $\pi(H(\ulcorner t_1\urcorner))$ is derivable from the set of terms $\{\pi(H(\mathrm{aenc}(t_1,t_2))),\pi(H(\ulcorner\mathrm{priv}(t_2)\urcorner))\}$. Consider two cases:

- $\exists u \in \mathring{\mathrm{Sub}}(S)$ such that $\ulcorner\mathrm{aenc}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{aenc}(t_1,t_2))) = \mathrm{aenc}(\pi(H(t_1)),\pi(H(t_2)))$, and then $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\{\mathrm{aenc}(\pi(H(t_1)),\pi(H(t_2))),\ulcorner\mathrm{priv}(\pi(H(t_2)))\urcorner\})$. On the other hand, we have $\pi(H(\ulcorner\mathrm{priv}(t_2)\urcorner)) = \pi(H(\mathrm{priv}(t_2))) = \pi(\{\mathrm{priv}(\pi(H(t_2)))\}) = \ulcorner\mathrm{priv}(\pi(H(t_2)))\urcorner$.
- $\nexists u \in \mathrm{QSub}(S)$ such that $\ulcorner\mathrm{aenc}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{aenc}(t_1,t_2))) = \pi(H(t_1)\cup H(t_2))$. Using Lemma 10, we have $\ulcorner H(t_1)\cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{aenc}(t_1,t_2)))\})$, thus $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{aenc}(t_1,t_2)))\})$. And then, by Lemma 10 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 14 and Lemma 13, $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\pi(H(\mathrm{aenc}(t_1,t_2))))$.

□

Using Propositions 2 and 3 we will show that transformation $\pi(H(\cdot))$ preserves the property of a substitution to be a model.

Proposition 4. *Given a normalized constraint system $\mathcal{S}$ and its normalized model σ. Then substitution $\pi(\mathrm{H}(\sigma))$ also satisfies $\mathcal{S}$.*

Proof. Suppose, without loss of generality, $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\dots,n}$. Let us take any constraint $(E \rhd t) \in \mathcal{S}$. As σ is a model of $\mathcal{S}$, there exists a derivation $D = A_0, \dots, A_k$ such that $A_0 = \ulcorner E\sigma \urcorner$ and $\ulcorner t\sigma \urcorner \in A_k$.

By Proposition 3 and Lemma 14 we can easily prove that if $k > 0$, $\pi(H(A_j)) \subseteq \mathrm{Der}(\pi(H(A_{j-1})))$, $j = 1, \dots, k$. Then, applying transitivity of $\mathrm{Der}(\cdot)$ (Lemma 14) k times, we have that $\pi(H(A_k)) \subseteq \mathrm{Der}(\pi(H(A_0)))$. In the case where $k = 0$, the statement $\pi(H(A_k)) \subseteq \mathrm{Der}(\pi(H(A_0)))$ is also true.

Using Proposition 2 we have that $\pi(H(A_0)) = \pi(H(E\sigma)) = \ulcorner E\,\pi(\mathrm{H}(\sigma))\urcorner$, as $E \subseteq \mathrm{Sub}(\mathcal{S})$. The same for t: $\pi(H(t\sigma)) = \ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner$, and as $\ulcorner t\sigma \urcorner \in A_k$, we have $\ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner \in \pi(H(A_k))$. Therefore, we have that $\ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner \in \pi(H(A_k)) \subseteq \mathrm{Der}(\pi(H(A_0))) = \mathrm{Der}(\ulcorner E\,\pi(\mathrm{H}(\sigma))\urcorner)$, that means $\pi(\mathrm{H}(\sigma))$ satisfies any constraint of $\mathcal{S}$. □

Till the end of subsection we will study a useful property of $\pi(\mathrm{H}(\sigma))$. Proposition 5 and its corollary show that if a constraint system has a normalized model σ which maps different variables to different values, then there exists another normalized model $\pi(\mathrm{H}(\sigma))$ that maps any variable of its domain to an ACI-set of some non-variable subterms of the constraint system instantiated by itself and some private keys built with atoms of the constraint system. This kind of model will be called *conservative*.

Lemma 15. Given a normalized substitution σ and normalized term u. If $\ulcorner u\sigma \urcorner = \mathrm{bin}(p, q)$ and $u \notin \mathcal{X}$ and $x\sigma \neq y\sigma$ for all $x \neq y$ then there exists $s \in \mathrm{Sub}(u)$ such that $s = \mathrm{bin}(p', q')$ and $\ulcorner s\sigma \urcorner = \mathrm{bin}(p, q)$. The same is true in the case of $\ulcorner u\sigma \urcorner = \mathrm{priv}(p)$.

Proof. As $u = \ulcorner u \urcorner$ and $\ulcorner u\sigma \urcorner = \mathrm{bin}(p, q)$, we have:

- u not in form of $\cdot(L)$. Then, as $u \notin \mathcal{X}$ and $\ulcorner u\sigma \urcorner = \mathrm{bin}(p, q)$, we have $u = \mathrm{bin}(p', q')$ (where $\ulcorner p'\sigma \urcorner = p$ and $\ulcorner q'\sigma \urcorner = q$). Then we can choose $s = \mathrm{bin}(p', q') = u \in \mathrm{Sub}(u)$.
- $u = \cdot(t_1, \dots, t_m)$, $\forall i$ $\mathrm{root}(t_i) \neq \cdot$ and $m > 1$ since $u = \ulcorner u \urcorner$. Then for all i, t_i is either a variable, or $\mathrm{bin}(p'_i, q'_i)$. But as $x\sigma \neq y\sigma$ for all $x \neq y$ and as σ is normalized, we can claim that the set $\{t_1, \dots, t_m\}$ contains at most one variable ($\exists j : t_j \in \mathcal{X} \implies \forall k \neq j, t_k \notin \mathcal{X}$). Since $\ulcorner u\sigma \urcorner = \mathrm{bin}(p, q)$ we have $\forall i, j$ $\ulcorner t_i\sigma \urcorner = \ulcorner t_j\sigma \urcorner$, and then, as $m > 1$, there exists i such that $t_i = \mathrm{bin}(p'_i, q'_i)$. Then, by definition of normalization function and from $\ulcorner u\sigma \urcorner = \mathrm{bin}(p, q)$ we have that $\ulcorner \mathrm{elems}(u\sigma)\urcorner = \{\mathrm{bin}(p, q)\}$. Thus, $\ulcorner \mathrm{bin}(p'_i, q'_i)\,\sigma \urcorner = \mathrm{bin}(p, q)$, i.e. we can choose $s = t_i$, as $t_i \in \mathrm{Sub}(u)$ and $t_i = \mathrm{bin}(p'_i, q'_i)$.

The other case (priv) can be proved similarly. □

Definition 3.14. A substitution σ is a *conservative model* of a constraint system $\mathcal{S}$, iff

(1) σ is normalized;
(2) σ is a model of $\mathcal{S}$;
(3) For any $x \in \mathrm{dom}(\sigma)$ there exist $k \in \mathbb{N}$ and $s_1, \dots, s_k \in \mathring{\mathrm{Sub}}(S) \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$ such that $x\sigma = \pi(\{s_1\sigma, \dots, s_k\sigma\})$ and $s_i \neq s_j$, if $i \neq j$ and $\mathrm{root}(s_i) \neq \cdot$ for all i.

Proposition 5. *Assume we are given a normalized constraint system $\mathcal{S}$ and its normalized model σ such that $\forall x, y \in \mathrm{dom}(\sigma), x \neq y \implies x\sigma \neq y\sigma$. Then $\pi(\mathrm{H}(\sigma))$ is a conservative model of $\mathcal{S}$.*

Proof. From Proposition 4 we obtain that $\pi(\mathrm{H}(\sigma))$ is a normalized solution of $\mathcal{S}$. By definition, $x\,\pi(\mathrm{H}(\sigma)) = \pi(H(x\sigma))$. Let us take any $s \in H(x\sigma)$ (note that s is a ground term). Then, by definition of $H(\cdot)$ we have:

- either $s \in \mathcal{A}$. Then, by definition of $H(\cdot)$, $s \in (\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))$. Thus, $s\,\pi(\mathrm{H}(\sigma)) = s$, $s \in \mathring{\mathrm{Sub}}(S)$, $s \neq \cdot(L)$;
- or $s = \mathrm{bin}(\pi(H(t_1)), \pi(H(t_2)))$ and $\exists u \in \mathring{\mathrm{Sub}}(S)$ such that $\ulcorner u\sigma \urcorner = \ulcorner \mathrm{bin}(t_1, t_2) \urcorner = \mathrm{bin}(\ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner)$. As all conditions of Lemma 15 are satisfied, $\exists v \in \mathrm{Sub}(u)$ such that $\ulcorner v\sigma \urcorner = \mathrm{bin}(\ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner)$ and $v = \mathrm{bin}(p, q)$ and as $u \in \mathring{\mathrm{Sub}}(S)$ then $v \in \mathring{\mathrm{Sub}}(S)$. By Proposition 2, $\ulcorner v\,\pi(\mathrm{H}(\sigma)) \urcorner = \pi(H(v\sigma)) = \pi(H(\ulcorner v\sigma \urcorner)) = \pi(H(\mathrm{bin}(t_1, t_2))) = \pi(\{\mathrm{bin}(\pi(H(t_1)), \pi(H(t_2)))\}) = \mathrm{bin}(\pi(H(t_1)), \pi(H(t_2))) = s$. That means, $\exists v \in \mathring{\mathrm{Sub}}(S), v \neq \cdot(L)$ such that $s = \ulcorner v\,\pi(\mathrm{H}(\sigma)) \urcorner$.
- or $s = \mathrm{priv}(\pi(H(t_1)))$. In this case, as s is ground, $\pi(H(t_1))$ must be an atom (we recall that $\pi(H(\cdot))$ transforms valid terms to valid ones), moreover, by definition of $H(\cdot)$, this atom is from $(\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))$. Therefore, $s = \mathrm{priv}(a)$, where $a \in \mathcal{A} \cap \mathrm{Sub}(\mathcal{S})$ (and of course, $s \neq \cdot(L)$).

Thus, for all $s \in H(x\sigma)$, there exists $v \in \mathrm{Sub}(\mathcal{S}) \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A}) \setminus \mathcal{X}$ such that $s = \ulcorner v\,\pi(\mathrm{H}(\sigma)) \urcorner$. Therefore, as $x\,\pi(\mathrm{H}(\sigma)) = \pi(H(x\sigma))$, we have that $x\,\pi(\mathrm{H}(\sigma)) = \pi(\{\ulcorner s_1\,\pi(\mathrm{H}(\sigma)) \urcorner, \ldots, \ulcorner s_k\,\pi(\mathrm{H}(\sigma)) \urcorner\}) = \pi(\{s_1\,\pi(\mathrm{H}(\sigma)), \ldots, s_k\,\pi(\mathrm{H}(\sigma))\})$, where $s_1, \ldots, s_k \in \mathring{\mathrm{Sub}}(\mathcal{S}) \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$ and $s_i \neq \cdot(L), \forall 1 \leq i \leq k$. That proves the proposition. □

Corollary 6. *Given a normalized constraint system $\mathcal{S}$ and its normalized model σ' such that $x \neq y \implies x\sigma' \neq y\sigma'$. Then $\mathcal{S}$ admits a conservative solution.*

3.5. Bounds on conservative solutions

To get a decidability result, we first show an upper bound on the size of a conservative model. Then we reduce any satisfiable constraint system to another satisfiable one that admits a conservative model (this reduction is, in fact, using one name for the variables on which some preliminary fixed model returns equal values). Moreover the considered conservative model of the obtained constraint system can be easily extended to a conservative model (of the same size) of the initial constraint system. We also show that the reduced constraint system has not a larger size than the original one. This means that the original constraint system has a model which is bounded with regard to the size of the constraint system. Thus, we prove the existence of a model with bounded size for any satisfiable constraint system.

Lemma 16. For any term t and substitution σ, $\forall s \in \mathrm{Sub}(t)\ \ |\mathrm{Sub}(\ulcorner t\sigma \urcorner)| \geq |\mathrm{Sub}(\ulcorner s\sigma \urcorner)|$.

Proof. From Lemma 5 we can easily obtain $\mathrm{Sub}(s\sigma) \subseteq \mathrm{Sub}(t\sigma)$. Then we need to prove that $|\mathrm{Sub}(\ulcorner p \urcorner)| \leq |\mathrm{Sub}(\ulcorner q \urcorner)|$, if $\mathrm{Sub}(p) \subseteq \mathrm{Sub}(q)$. The proof is mainly based on the fact that if $\mathrm{Sub}(p) \subseteq \mathrm{Sub}(q)$ then $\mathrm{Sub}(\ulcorner p \urcorner) \setminus \{\ulcorner p \urcorner\} \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$. Let us consider several cases.

- If $p = q$ then the statement is trivial.
- If there exists $v \in \mathrm{Sub}(q)$ such that $v = \mathrm{bin}(p, p')$ or $v = \mathrm{bin}(p', p)$ or $v = \mathrm{priv}(p)$. Then note that by definition of the normalization, since $\mathrm{root}(v) \neq \cdot$, we have $\ulcorner v \urcorner \in \mathrm{Sub}(\ulcorner q \urcorner)$. Then (without loss of generality we consider only case $v = \mathrm{bin}(p, p')$) $\ulcorner v \urcorner = \mathrm{bin}(\ulcorner p \urcorner, \ulcorner p' \urcorner)$ and thus $\ulcorner p \urcorner \in \mathrm{Sub}(\ulcorner v \urcorner)$. Therefore, (since $\ulcorner v \urcorner \in \mathrm{Sub}(\ulcorner q \urcorner)$) $\ulcorner p \urcorner \in \mathrm{Sub}(\ulcorner q \urcorner)$ and then $\mathrm{Sub}(\ulcorner p \urcorner) \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$. From this follows $|\mathrm{Sub}(\ulcorner p \urcorner)| \leq |\mathrm{Sub}(\ulcorner q \urcorner)|$.

- Otherwise, there exists $v \in \mathrm{QSub}(q)$ such that $v = \cdot(L)$ and $\mathrm{elems}(p) \subseteq \mathrm{elems}(v)$ (note that $v \neq p$, otherwise we are in the one of the two cases above). From Lemma 2 we have $\mathrm{elems}(\ulcorner p \urcorner) \subseteq \mathrm{elems}(\ulcorner v \urcorner)$. Note that by definition of the normalization and since $v \in \mathrm{QSub}(q)$ (and thus $v = q$ or there exists $v' \in \mathrm{Sub}(q)$ such that $v' = \mathrm{bin}(v, p')$ or $v' = \mathrm{bin}(p', v)$ or $v' = \mathrm{priv}(v)$) we have that $\ulcorner v \urcorner \in \mathrm{Sub}(\ulcorner q \urcorner)$, moreover, $\mathrm{Sub}(\ulcorner v \urcorner) \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$. Then from Lemma 6 we have $\mathrm{elems}(\ulcorner v \urcorner) \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$. Thus, $\mathrm{elems}(\ulcorner p \urcorner) \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$, consequently, $\mathrm{Sub}(\mathrm{elems}(\ulcorner p \urcorner)) \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$. Now, if $\mathrm{elems}(\ulcorner p \urcorner) = \{\ulcorner p \urcorner\}$, then the statement becomes trivial. Otherwise, $\ulcorner p \urcorner = \cdot(\mathrm{elems}(\ulcorner p \urcorner))$, and then $\mathrm{Sub}(\ulcorner p \urcorner) = \ulcorner p \urcorner \cup \mathrm{Sub}(\mathrm{elems}(\ulcorner p \urcorner))$. Since we have already shown that $\mathrm{Sub}(\mathrm{elems}(\ulcorner p \urcorner)) \subseteq \mathrm{Sub}(\ulcorner q \urcorner)$, to prove the statement it is enough to show that there exists $p' \in \mathrm{Sub}(\ulcorner q \urcorner)$ such that $p' \notin \mathrm{Sub}(\mathrm{elems}(\ulcorner p \urcorner))$. For such value we can choose $p' = \ulcorner v \urcorner$ ($\ulcorner v \urcorner$ cannot be in $\mathrm{Sub}(\mathrm{elems}(\ulcorner p \urcorner))$, since $\mathrm{elems}(\ulcorner p \urcorner) \subseteq \mathrm{elems}(\ulcorner v \urcorner) \subseteq \mathrm{Sub}(\ulcorner v \urcorner)$ and in the current case $\mathrm{root}(\ulcorner v \urcorner) = \cdot$).
  □

Lemma 17. Given a normalized constraint system $\mathcal{S}$ and its conservative model σ. Then $\forall x \in \mathrm{Vars}(\mathcal{S}), \mathrm{Sub}(x\sigma) \subseteq \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$.

Proof. Given a ground substitution σ, let us define a strict total order on variables: $x \sqsubset y \iff (|\mathrm{Sub}(x\sigma)| < |\mathrm{Sub}(y\sigma)|) \vee (|\mathrm{Sub}(x\sigma)| = |\mathrm{Sub}(y\sigma)| \wedge x \prec y)$.

By Proposition 5 for any $x \in \mathrm{Vars}(\mathcal{S})$ we have $x\sigma = \pi(\{s_1^x\sigma, \ldots, s_{k^x}^x\sigma\})$, where $s_i^x \in (\mathrm{Sub}(\mathcal{S}) \setminus \mathcal{X}) \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$ and $s_i^x \neq \cdot(L)$. Moreover, s_i^x are normalized (as $\mathcal{S}$ is normalized).

Let us show that if $y \in \mathrm{Vars}(s_i^x)$ for some i, then $y \sqsubset x$. Suppose that $y \in \mathrm{Vars}(s_i^x)$ and $x \sqsubset y$. Then $|\mathrm{Sub}(x\sigma)| = |\mathrm{Sub}(\pi(\{s_1^x\sigma, \ldots, s_{k^x}^x\sigma\}))| = |\mathrm{Sub}(\ulcorner \cdot(s_1^x\sigma, \ldots, s_{k^x}^x\sigma) \urcorner)| \geq$ (by Lemma 16) $\geq |\mathrm{Sub}(\ulcorner s_i^x\sigma \urcorner)| > |\mathrm{Sub}(\ulcorner y\sigma \urcorner)|$, because we know that $s_i^x = \mathrm{bin}(p, q)$ or $s_i^x = \mathrm{priv}(p)$ and $y \in \mathrm{Vars}(s_i^x)$ (for example, in the first case, $|\mathrm{Sub}(\ulcorner s_i^x\sigma \urcorner)| = |\mathrm{Sub}(\mathrm{bin}(\ulcorner p\sigma \urcorner, \ulcorner q\sigma \urcorner))| = 1 + |\mathrm{Sub}(\{\ulcorner p\sigma \urcorner, \ulcorner q\sigma \urcorner\})|$ and as $y \in \mathrm{Vars}(\{p, q\})$, using Lemma 16, we get $|\mathrm{Sub}(\ulcorner s_i^x\sigma \urcorner)| \geq 1 + |\mathrm{Sub}(\ulcorner y\sigma \urcorner)|$). Since $|\mathrm{Sub}(\ulcorner y\sigma \urcorner)| = |\mathrm{Sub}(y\sigma)|$ we have $y \sqsubset x$. Contradiction.

Now we show by induction the main property of this lemma.

- let $x = \min_{\sqsubset}(\mathrm{Vars}(\mathcal{S}))$. Then $x\sigma = \pi(\{s_1^x\sigma, \ldots, s_{k^x}^x\sigma\}) = \ulcorner \cdot(s_1^x\sigma, \ldots, s_{k^x}^x\sigma) \urcorner$ and all s_i^x are ground (as $\nexists y \sqsubset x$). Then $x\sigma = \ulcorner \cdot(s_1^x, \ldots, s_{k^x}^x) \urcorner$. Since s_i^x are normalized with non-$\cdot$ root, we have that $\mathrm{Sub}(x\sigma) = \{\ulcorner \cdot(s_1^x, \ldots, s_{k^x}^x) \urcorner\} \cup \mathrm{Sub}(s_1^x) \cup \cdots \cup \mathrm{Sub}(s_{k^x}^x) \subseteq \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))$, as $\forall s \in \mathrm{Sub}(s_i^x)$, $s \in \mathcal{T}_g$ and $s \in \mathrm{Sub}(\mathcal{S})$ or $s = \mathrm{priv}(a)$ or $s = a$, where $a \in \mathrm{Sub}(\mathcal{S}) \cap \mathcal{A}$, therefore $s = \ulcorner s \urcorner = s\sigma \in \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$ and $\ulcorner \cdot(s_1^x, \ldots, s_{k^x}^x) \urcorner = x\sigma \in \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner$.
- Suppose, for all $z \sqsubset y$, $\mathrm{Sub}(z\sigma) \subseteq \ulcorner \mathrm{Sub}(\mathcal{S}\sigma) \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$.
- Show that $\mathrm{Sub}(y\sigma) \subseteq \mathrm{Sub}(\mathcal{S}\sigma) \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$.
  We know that $y\sigma = \pi(\{s_1^y\sigma, \ldots, s_{k^y}^y\sigma\}) = \ulcorner \cdot(s_1^y\sigma, \ldots, s_{k^y}^y\sigma) \urcorner$ and $\forall z \in \mathrm{Vars}(s_i^y), z \sqsubset y$. Then we have $\mathrm{Sub}(y\sigma) = \{y\sigma\} \cup \mathrm{Sub}(\ulcorner s_1^y\sigma \urcorner) \cup \cdots \cup \mathrm{Sub}(\ulcorner s_{k^y}^y\sigma \urcorner)$. We know that $y\sigma \in \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner$. Let us show that $\mathrm{Sub}(\ulcorner s_i^y\sigma \urcorner) \subseteq \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$. By Theorems 5 and 9 we have $\mathrm{Sub}(\ulcorner s_i^y\sigma \urcorner) \subseteq \ulcorner \mathrm{Sub}(s_i^y\sigma) \urcorner = \ulcorner \mathrm{Sub}(s_i^y)\,\sigma \cup \mathrm{Sub}(\mathrm{Vars}(s_i^y)\,\sigma) \urcorner = \ulcorner \mathrm{Sub}(s_i^y)\,\sigma \urcorner \cup \mathrm{Sub}(\mathrm{Vars}(s_i^y)\,\sigma)$. As $s_i^y \in \mathrm{Sub}(\mathcal{S}) \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$, we can see that $\ulcorner \mathrm{Sub}(s_i^y)\,\sigma \urcorner \subseteq \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$; then from the induction hypothesis and the statement proved above we have $\mathrm{Sub}(\mathrm{Vars}(s_i^y)\,\sigma) \subseteq \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$. Thus, $\mathrm{Sub}(y\sigma) \subseteq \ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{Sub}(\mathcal{S}) \cap \mathcal{A})$.
  □

Proposition 7. *For a normalized constraint system $\mathcal{S}$ that has a conservative model σ, $\forall x \in \mathrm{Vars}(\mathcal{S}), |\mathrm{Sub}(x\sigma)| \leq 2 \times |\mathrm{Sub}(\mathcal{S})|$.*

Proof. As $|\ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner| \leq |\mathrm{Sub}(\mathcal{S})\,\sigma| \leq |\mathrm{Sub}(\mathcal{S})|$, we have that $|\mathrm{Sub}(x\sigma)| \leq |\ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))| \leq |\ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner| + |\mathrm{priv}(\mathcal{A} \cap \mathrm{Sub}(\mathcal{S}))| \leq$ (as $|\ulcorner T \urcorner| \leq |T|$ and $|T\sigma| \leq |T|$) $\leq |\mathrm{Sub}(\mathcal{S})| + |\mathcal{A} \cap \mathrm{Sub}(\mathcal{S})| \leq 2 \times |\mathrm{Sub}(\mathcal{S})|$; thus, $|\mathrm{Sub}(x\sigma)| \leq 2 \times |\mathrm{Sub}(\mathcal{S})|$. □

From this proposition and Corollary 6 we obtain the existence of a bounded model for any normalized constraint system that has a model mapping different variables to different values. We will reduce an arbitrary constraint system to the already studied case. The target properties are stated in Proposition 8 and Corollary 9.

Lemma 18. Given any constraint system $\mathcal{S}$ and any substitution θ such that $\mathrm{dom}(\theta) = \mathrm{Vars}(\mathcal{S})$ and $\mathrm{img}(\theta) \subseteq \mathrm{dom}(\theta)$, then $|\mathrm{Sub}(\mathcal{S}\theta)| \leq |\mathrm{Sub}(\mathcal{S})|$.

Proof. By Lemma 5, $|\mathrm{Sub}(\mathcal{S}\theta)| = |\mathrm{Sub}(\mathcal{S})\,\theta \cup \mathrm{Sub}(\mathrm{Vars}(\mathcal{S})\,\theta)|$, but $\mathrm{Vars}(\mathcal{S})\,\theta \subseteq \mathrm{dom}(\theta) = \mathrm{Vars}(\mathcal{S})$ ($\mathrm{Vars}(\mathcal{S})\,\theta$ consists only of variables), and then $\mathrm{Sub}(\mathrm{Vars}(\mathcal{S})\,\theta) = \mathrm{Vars}(\mathcal{S})\,\theta$. As $\mathrm{Vars}(\mathcal{S}) \subseteq \mathrm{Sub}(\mathcal{S})$, we have $\mathrm{Sub}(\mathcal{S})\,\theta \cup \mathrm{Sub}(\mathrm{Vars}(\mathcal{S})\,\theta) = \mathrm{Sub}(\mathcal{S})\,\theta$. Thus, $|\mathrm{Sub}(\mathcal{S}\theta)| = |\mathrm{Sub}(\mathcal{S})\,\theta| \leq |\mathrm{Sub}(\mathcal{S})|$. □

Definition 3.15. Let σ and δ be two substitutions. Then $\sigma[\delta]$ is a substitution such that $\mathrm{dom}(\sigma[\delta]) = \mathrm{dom}(\delta)$ and for all $x \in \mathrm{dom}(\sigma[\delta])$, $x\sigma[\delta] = (x\delta)\sigma$.

Lemma 19. Let θ and σ be two substitutions such that $\mathrm{img}(\theta) = \mathrm{dom}(\sigma)$, $\mathrm{dom}(\sigma) \subseteq \mathrm{dom}(\theta)$. Then, for any term t, $(t\theta)\sigma = t\sigma[\theta]$.

Proof. When we apply θ to t, every variable x of t such that $x \in \mathrm{dom}(\theta)$ is replaced by $x\theta$; then we apply σ to $t\theta$: every variable y of $t\theta$ is replaced by $y\sigma$, thus, every variable x from $\mathrm{dom}(\theta)$ will be replaced by $(x\theta)\sigma$ (as $\mathrm{img}(\theta) = \mathrm{dom}(\sigma)$); and no other variables will be replaced (as $\mathrm{dom}(\sigma) \subseteq \mathrm{dom}(\theta)$). This corresponds with the definition of $\sigma[\theta]$. □

Proposition 8. *Given a satisfiable constraint system $\mathcal{S}$ there exists a model σ of $\mathcal{S}$ such that $\forall x \in \mathrm{dom}(\sigma), |\mathrm{Sub}(x\sigma)| \leq 2 \times |\mathrm{Sub}(\ulcorner \mathcal{S} \urcorner)|$*

Proof. Given a normalized model σ' of $\mathcal{S}$ we build a substitution θ that maps different variables whose σ'-values are equal to a single new variable. In this way we obtain a new constraint system $\ulcorner \mathcal{S}\theta \urcorner$ and its normalized model. Then we apply Corollary 6 and get a conservative model σ'' of $\ulcorner \mathcal{S}\theta \urcorner$. By applying Proposition 7 we get a bound on the size for this model. On the other hand , we use Lemma 19 to show that $\sigma''[\theta]$ is a model of $\ulcorner \mathcal{S} \urcorner$. Then, using the obtained bound and Lemma 18, we show the existence of a model with the desired property. The detailed proof is given in Appendix B.6 □

Corollary 9. *A constraint system $\mathcal{S}$ is satisfiable iff there exists a normalized model of $\mathcal{S}$ defined on $\mathrm{Vars}(\mathcal{S})$ which maps a variable to a ground term in $\mathcal{T}(\mathcal{A} \cap \mathrm{QSub}(\ulcorner \mathcal{S} \urcorner), \emptyset)$ of size $\leq 2 \times |\mathrm{Sub}(\mathcal{S})|$.*

Using this result we get Algorithm 1 for solving constraint systems:

Algorithm 1: Solving constraint systems

Input: A constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\dots,n}$
Output: A model σ, if it exists else $\bot$

1 **Guess** for every variable x of $\mathcal{S}$ a value for $x\sigma$ of size $\leq 2 \times |\mathrm{Sub}\,(\mathcal{S})|$;
2 **if** *σ satisfies $E_i \rhd t_i$ for all $i = 1, \dots, n$* **then return** σ **else return** $\bot$

Proposition 10. *Algorithm 1 is correct.*

Proof. Let σ be an output of Algorithm 1. Then σ is a ground substitution and σ satisfies all constraints from $\mathcal{S}'$ and therefore, satisfies all constraints from $\mathcal{S}$. This means, σ is a model of $\mathcal{S}$. □

Proposition 11. *Algorithm 1 is complete.*

Proof. Suppose $\mathcal{S}$ is satisfiable. Then, by Corollary 9, there exists a guess of values for every element of $\mathrm{Vars}\,(\mathcal{S})$ with size $\leq 2 \times |\mathrm{Sub}\,(\mathcal{S})|$ which defines a model σ of $\mathcal{S}$. Thus Algorithm 1 will return this σ. □

4. Complexity analysis

In this section we study the complexity of the proposed algorithm. First, we expose a representation of constraint systems to justify the selected measure for algorithm inputs. Then, we remark that the normalization algorithm is polynomial and we show that the derivability algorithm is polynomial too. As a consequence the algorithm for solving general constraint systems within the DY+ACI model is in *NP*. Since constraint solving is known to be *NP*-hard for well-formed constraints, we deduce it is *NP*-complete for general constraints.

To determine the algorithm complexity we first define a reasonable measure for inputs. Remark that $|\mathrm{Sub}\,(\cdot)|$ does not approximate polynomially the number of bits needed to write its argument (a term, a set of terms, or a constraint system). As a measure for terms and sets of terms we will use $|\mathrm{Sub}\,(\cdot)| + |\mathbb{E}\,(\cdot)|$, where $\mathbb{E}\,(\cdot)$ is a set of edges in the DAG-representation of its argument. For the size of a system of constraints $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\dots,n}$ we will use $n \times |\mathrm{Sub}\,(\mathcal{S})| + |\mathbb{E}\,(\mathcal{S})|$. The justification is given below.

Definition 4.1. The *DAG-representation* of a constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\dots,n}$ is a tagged graph with labeled edges $\mathbb{G} = \langle \mathbb{V}, \mathbb{E}, \mathrm{tag} \rangle$ ($\mathbb{V}$ is a set of vertices and $\mathbb{E}$ is a set of edges; tag is a tagging function defined on $\mathbb{V}$) such that:

- there exists a bijection $f : \mathbb{V} \mapsto \mathrm{Sub}\,(\mathcal{S})$;
- $\forall v \in \mathbb{V}$ $\mathrm{tag}\,(v) = \langle s, m \rangle$, where
  - $s = \mathrm{root}\,(f(v))$;
  - m is a $2n$-bit integer, where $m[2i-1] = 1$ if and only if $f(v) \in E_i$ and $m[2i] = 1$ if and only if $f(v) = t_i$.
- $v_1 \xrightarrow{1} v_2 \in \mathbb{E}$ if and only if $\exists p \in \mathcal{T} : (\exists\, \mathrm{bin} : f(v_1) = \mathrm{bin}\,(f(v_2), p)) \vee f(v_1) = \mathrm{priv}\,(f(v_2))$;
- $v_1 \xrightarrow{2} v_2 \in \mathbb{E}$ if and only if $\exists p \in \mathcal{T} : \exists\, \mathrm{bin} : f(v_1) = \mathrm{bin}\,(p, f(v_2))$;
- $v_1 \xrightarrow{i} v_2 \in \mathbb{E}$ if and only if $f(v_1) = \cdot\,(L) \wedge L[i] = f(v_2)$;

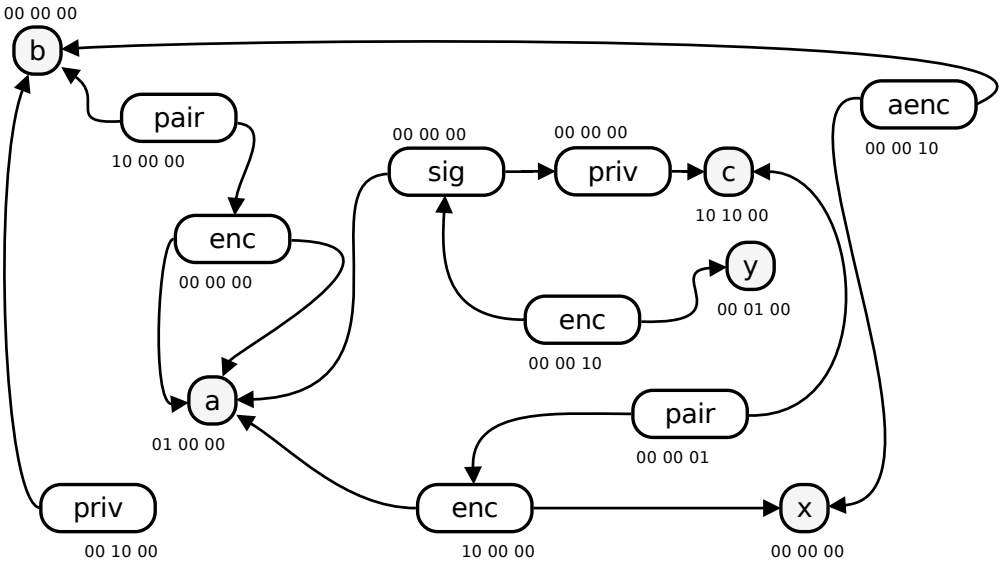


Figure 4. DAG-representation of constraint system $\mathcal{S}$

Example 7. A constraint system

$$\mathcal{S} = \begin{cases} \{\mathrm{enc}(a, x), \mathrm{pair}(b, \mathrm{enc}(a, a)), c\} \rhd a \\ \{\mathrm{priv}(b), c\} \rhd y \\ \{\mathrm{enc}(\mathrm{sig}(a, \mathrm{priv}(c)), y), \mathrm{aenc}(x, b)\} \rhd \mathrm{pair}(\mathrm{enc}(a, x), c) \end{cases}$$

will be represented as shown [3] in Figure 4. The nodes of this graph represent the elements from $\mathrm{Sub}(\mathcal{S})$ by indicating their root symbols (first part of their tags) and pointers to their children.

The given representation takes less than $P(n \times |\mathbb{V}(\mathcal{S})| + |\mathbb{E}(\mathcal{S})|)$ bits, where n is the number of constraints, $\mathbb{V}(\cdot)$ is the set of nodes and $\mathbb{E}(\cdot)$ is the set of edges in the DAG-representation, and P is some polynomial with non-negative coefficients. As we have a bijection between $\mathbb{V}(\mathcal{S})$ and $\mathrm{Sub}(\mathcal{S})$, we obtain $|\mathbb{V}(\mathcal{S})| = |\mathrm{Sub}(\mathcal{S})|$. Therefore we will estimate the complexity of our algorithm by taking $n \times |\mathrm{Sub}(\mathcal{S})| + |\mathbb{E}(\mathcal{S})|$ as the measure of the constraint system $\mathcal{S}$.

The DAG-representation of a term t has a structure similar to the one of a constraint system except that the second part of the tagging function is dropped: we need only $\mathrm{root}(f(v))$ as a node's tag. The size of this representation will be polynomially bounded by $|\mathrm{Sub}(t)| + |\mathbb{E}(t)|$. Thus we give the following definition:

Definition 4.2. The *measure* of a term t is: $\mathrm{measure}(t) = |\mathrm{Sub}(t)| + |\mathbb{E}(t)|$. The *measure* of a constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ is: $\mathrm{measure}(\mathcal{S}) = n \times |\mathrm{Sub}(\mathcal{S})| + |\mathbb{E}(\mathcal{S})|$.

[3] Label "1" (resp."2") of an edge is represented by a left (resp. right) side of its source node

We remark that the given measure is not the exact number of bits needed to write the argument down (this value involves a logarithm factor and would complexify the notation), but is a polynomial approximation of it which is sufficient for our purpose.

Note that for normalized terms and constraint systems the number of edges in their DAG-representation are polynomially bounded w.r.t. the number of vertices:

Lemma 20. For any normalized term t, $|\mathbb{E}(t)| < (|\mathrm{Sub}(t)|)^2$. For any normalized constraint system $\mathcal{S}$, $|\mathbb{E}(\mathcal{S})| < (|\mathrm{Sub}(\mathcal{S})|)^2$.

Proof. Since the term (resp. constraint system) is normalized, there are at most two edges between two nodes. A binary node has at most two outgoing edges per destination node, an unary node — one, and $\cdot$-node has at most one outgoing edge per destination node because of normalization. Therefore, as the graph is directed and acyclic, with at most two edges between two nodes, we have less than $|\mathrm{Sub}(x)| \times (|\mathrm{Sub}(x)| - 1)$ edges (where x is a term t or a constraint system $\mathcal{S}$). □

4.1. Satisfiability of a general DY+ACI constraint system is in NP

Lemma 21. The normalization of a term t can be done in polynomial time on measure (t). The same holds for a constraint system $\mathcal{S}$: normalization can be done in polynomial time on measure $(\mathcal{S})$.

Proof idea (for the case of terms). The algorithm of term normalization works bottom-up by flattening nested ACI-sets, sorting children of ACI-set nodes, merging duplicated nodes while removing unnecessary duplicating edges and removing nodes without incoming edges (except the root-node of t). □

Lemma 22. For a term t, $|\mathrm{Sub}(\ulcorner t \urcorner)| \leq |\mathrm{Sub}(t)|$; for a set of terms T, $|\mathrm{Sub}(\ulcorner T \urcorner)| \leq |\mathrm{Sub}(T)|$; for a constraint system $\mathcal{S}$, $|\mathrm{Sub}(\ulcorner \mathcal{S} \urcorner)| \leq |\mathrm{Sub}(\mathcal{S})|$.
The proof for the case of terms is given in Appendix B.5. □

Proposition 12. *Satisfiability of general DY+ACI constraint systems is in NP.*

Proof. Algorithm 1 returns a solution if it exists. We have to show that the verification of this solution is polynomial with regard to the input. To do this, we will normalize $\mathcal{S}\sigma$ and then apply the result described in § 3.3 for checking derivability. Using the fact that $|\mathrm{Sub}(x\sigma)| \leq 2 \times |\mathrm{Sub}(\mathcal{S})|$ and normalization is polynomial, checking the validity of terms in $\ulcorner \mathcal{S}\sigma \urcorner$ and the derivability, we can get an upper bound of the execution time that is polynomial on measure $(\mathcal{S})$. The details of the proof are given in Appendix B.3. □

On the other hand, we can reuse a technique presented in Rusinowitch and Turuani (2003) to show that the satisfiability of a constraint system is an *NP*-hard problem. The authors encode the 3-SAT problem into an insecurity problem of a single-session sequential protocol. Because the steps of the protocol are linearly ordered, finding an attack is reduced to the satisfiability problem of a single well-formed constraint system. Since the class of constraint systems we consider contains the case presented in this work, we can conclude to the *NP*-hardness of our problem. Thus,

Theorem 1. Satisfiability of general DY+ACI constraint systems is *NP*-complete.

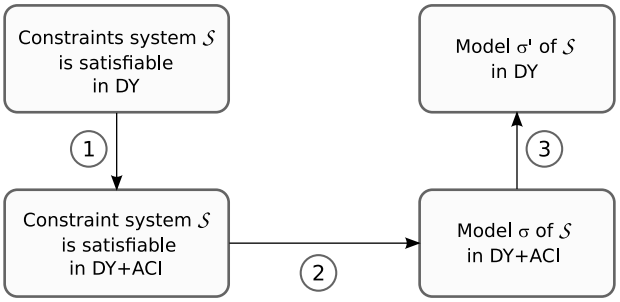


Figure 5. Proof Plan

5. Satisfiability of general DY constraint systems

The constraint solving algorithm for the DY+ACI theory can be adapted to the classical DY case. But we cannot apply it directly, as a derived solution may contain an ACI symbol. Thus, we need to do some more work to prove the decidability of the DY case. The scheme we follow to solve a constraint system within the DY deduction system is shown in Figure 5.

First, we show that if a constraint system is satisfiable within DY, then it is satisfiable within DY+ACI (Proposition 13). Second, as we know, we can find a model of a given constraint system within DY+ACI. Third, we transform the model obtained from previous step (which is in DY+ACI) in such a way, that the resulting substitution will be a model of the initial constraint system within DY (Theorem 2). The transformation δ to be used is simple: we replace any ACI list of terms $\cdot(t_1, \ldots, t_n)$ by nested pairs: $\text{pair}(t_1, \text{pair}(..., t_n))$. Note that this transformation has a linear complexity and the transformed model will have a DAG-size not more than twice bigger than the initial one. This keeps the complexity in *NP*, for the problem of satisfiability of general constraint system within the DY model.

Definition 5.1. We define a replacement $\delta(t) : \mathcal{T}_g \mapsto \mathcal{T}_g$ in the following way:

$$\delta(t) = \begin{cases} t, \text{ if } t \in \mathcal{X} \cup \mathcal{A}; \\ \text{bin}(\delta(p), \delta(q)), \text{ if } t = \text{bin}(p, q), \\ \text{priv}(\delta(p)), \text{ if } t = \text{priv}(p); \\ \delta(t_1), \text{ if } t = \cdot(t_1); \\ \text{pair}(\delta(t_1), \delta(\cdot(t_2, \ldots, t_m))), \text{ if } t = \cdot(t_1, \ldots, t_m), m > 1; \end{cases}$$

Definition 5.2. Given a substitution σ. Then $\delta(\sigma) = \{x \to \delta(x\sigma)\}_{x \in \text{dom}(\sigma)}$. For $T \subseteq \mathcal{T}_g$, $\delta(T) = \{\delta(t) : t \in T\}$.

To recall the Dolev-Yao deduction system (DY), see Table 1.

Definition 5.3. A constraint system $\mathcal{S}$ is *standard* if for all $s \in \text{Sub}(\mathcal{S})$ $\text{root}(s) \neq \cdot$. The definition is extended in natural way to terms, sets of terms and substitutions.

We can redefine the notion of derivation for Dolev-Yao deduction system in a natural way, and denote it as Der_{DY}.

Lemma 23. Any standard constraint system is normalized.

Lemma 24. Let t be a standard term and σ be a normalized substitution. Then $t\sigma$ is normalized.

Proposition 13. *If a standard constraint system $\mathcal{S}$ has a model σ within the DY deduction system, then $\mathcal{S}$ has a model within the DY+ACI deduction system.*

Proof. It is enough to consider the same model σ in DY+ACI. As $\mathcal{S}\sigma$ is normalized and as DY+ACI includes all the rules from DY, it is easy to show using the same derivation that proves σ to be a model in DY, that σ remains a model of $\mathcal{S}$ in DY+ACI. □

Next we show that we can build a model of a constraint system within DY from a model of this constraint system within DY+ACI.

Proposition 14. *For any DY+ACI rule $l_1, \ldots, l_k \to r$, if l_i are normalized for all $i = 1, \ldots, k$ then $\delta(r) \in \mathrm{Der}_{DY}(\{\delta(l_1), \ldots, \delta(l_k)\})$.*

Proof. Let us consider all possible rules:

(1) $t_1, t_2 \to \ulcorner \mathrm{bin}(t_1, t_2) \urcorner$ for bin $\neq$ sig.
As t_1 and t_2 are normalized, then $\ulcorner \mathrm{bin}(t_1, t_2) \urcorner = \mathrm{bin}(t_1, t_2)$. We can see, that $\delta(\mathrm{bin}(t_1, t_2)) = \mathrm{bin}(\delta(t_1), \delta(t_2)) \in \mathrm{Der}_{DY}(\{\delta(t_1), \delta(t_2)\})$.

(2) $t_1, \mathrm{priv}(t_2) \to \ulcorner \mathrm{sig}(t_1, \mathrm{priv}(t_2)) \urcorner$.
As t_1 and $\mathrm{priv}(t_2)$ are normalized, then $\ulcorner \mathrm{sig}(t_1, \mathrm{priv}(t_2)) \urcorner = \mathrm{sig}(t_1, \mathrm{priv}(t_2))$. We can see that $\delta(\mathrm{sig}(t_1, \mathrm{priv}(t_2))) = \mathrm{sig}(\delta(t_1), \delta(\mathrm{priv}(t_2))) = \mathrm{sig}(\delta(t_1), \mathrm{priv}(\delta(t_2))) \in \mathrm{Der}_{DY}(\{\delta(t_1), \mathrm{priv}(\delta(t_2))\})$, but $\mathrm{priv}(\delta(t_2)) = \delta(\mathrm{priv}(t_2))$.

(3) $t_1, \ldots, t_m \to \ulcorner \cdot(t_1, \ldots, t_m) \urcorner$.
The fact that $\mathrm{elems}(\ulcorner \cdot(t_1, \ldots, t_m) \urcorner) = \bigcup_{i=1,\ldots,m} \mathrm{elems}(t_i)$ follows from $t_i = \ulcorner t_i \urcorner$ (for all i) and Lemma 4.
We can (DY)-derive from $\{\delta(t_i)\}$ any term in $\delta(\mathrm{elems}(t_i))$ trivially if $t_i \neq \cdot(L)$ and by applying rules $\mathrm{pair}(s_1, s_2) \xrightarrow{DY} s_1$ and $\mathrm{pair}(s_1, s_2) \xrightarrow{DY} s_2$ otherwise (by induction on the size of t_i).
One can observe, that $\delta(t)$ is a pairing (composition of $\mathrm{pair}(\cdot, \cdot)$ operator with itself) of $\delta(\mathrm{elems}(t))$ (by definition of $\delta(\cdot)$ and normalization function). And then, as $\delta(t)$ is limited in size, we can (DY)-derive $\delta(t)$ from $\delta(\mathrm{elems}(t))$ by iterative use of rule $s_1, s_2 \xrightarrow{DY} \mathrm{pair}(s_1, s_2)$, if needed.
Thus, first we can derive $\delta(\mathrm{elems}(t_i))$ for all i, and then rebuild (derive with composition rules) $\delta(\ulcorner \cdot(t_1, \ldots, t_m) \urcorner)$.

(4) $\mathrm{enc}(t_1, t_2), \ulcorner t_2 \urcorner \to \ulcorner t_1 \urcorner$.
Since $\mathrm{enc}(t_1, t_2)$ is normalized, we have $t_1 = \ulcorner t_1 \urcorner$ and $t_2 = \ulcorner t_2 \urcorner$. Thus, $\delta(t_1) \in \mathrm{Der}_{DY}(\{\mathrm{enc}(\delta(t_1), \delta(t_2)), \delta(t_2)\})$ and this is what we need, as $\delta(\mathrm{enc}(t_1, t_2)) = \mathrm{enc}(\delta(t_1), \delta(t_2))$.

(5) $\mathrm{pair}(t_1, t_2) \to \ulcorner t_1 \urcorner$. Similar case.

(6) $\mathrm{pair}(t_1, t_2) \to \ulcorner t_2 \urcorner$. Similar case.

(7) $\mathrm{aenc}(t_1, t_2), \ulcorner \mathrm{priv}(t_2) \urcorner \to \ulcorner t_1 \urcorner$. Similar case. Note, that $\delta(\mathrm{priv}(t_2)) = \mathrm{priv}(\delta(t_2))$

(8) $\cdot(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$.
As said above, $\delta(\mathrm{elems}(\cdot(t_1, \ldots, t_m))) \subseteq \mathrm{Der}_{DY}(\{\delta(\cdot(t_1, \ldots, t_m))\})$; and as $\delta(\mathrm{elems}(t_i)) \subseteq \delta(\mathrm{elems}(\cdot(t_1, \ldots, t_m)))$, we can (DY)-derive (by composition rules) $\delta(t_i)$ from $\delta(\mathrm{elems}(t_i))$.

□

Proposition 15. *Given a standard constraint system $\mathcal{S}$ and a normalized model σ of $\mathcal{S}$ in DY+ACI then, for any $t \in \mathrm{Sub}(\mathcal{S})$ we have $\delta(t\sigma) = t\delta(\sigma)$.*

Proof. The proof is by induction as in Proposition 2.
- Let $|\mathrm{Sub}(t)| = 1$. Then either $t \in \mathcal{A}$ or $t \in \mathcal{X}$. Both are trivial cases.
- Assume that for some $k \geq 1$ if $|\mathrm{Sub}(t)| \leq k$, then $\delta(t\sigma) = t\delta(\sigma)$.
- Show, that for t such that $|\mathrm{Sub}(t)| \geq k+1$, where $t = \mathrm{bin}(p,q)$ or $t = \mathrm{priv}(p)$ and $|\mathrm{Sub}(p)| \leq k$ and $|\mathrm{Sub}(q)| \leq k$, statement $\delta(t\sigma) = t\delta(\sigma)$ is still true. We have:
  · either $t = \mathrm{bin}(p,q)$. As $\delta(\mathrm{bin}(p,q)\sigma) = \delta(\mathrm{bin}(p\sigma, q\sigma)) = \mathrm{bin}(\delta(p\sigma), \delta(q\sigma)) = \mathrm{bin}(p\delta(\sigma), q\delta(\sigma)) = \mathrm{bin}(p,q)\delta(\sigma)$.
  · or $t = \mathrm{priv}(p)$. In this case the proof can be done by analogy with previous one.
  Remark: as $\mathcal{S}$ is standard, $t \neq \cdot(L)$.
  □

Theorem 2. Let $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ be a standard constraint system and σ be a normalized model in DY+ACI. Then $\delta(\sigma)$ is a model in DY of $\mathcal{S}$.

Proof. Let $E \rhd t$ be any element of $\mathcal{S}$. As σ is a model of $\mathcal{S}$, then $\ulcorner t\sigma \urcorner \in \mathrm{Der}(\ulcorner E\sigma \urcorner)$. As σ is normalized and $\mathcal{S}$ is standard, using Lemma 24 we have $\ulcorner t\sigma \urcorner = t\sigma$ and $\ulcorner E\sigma \urcorner = E\sigma$. Then, $t\sigma \in \mathrm{Der}(E\sigma)$. That means there exists a DY+ACI derivation $D = \{A_0, \ldots, A_k\}$ such that $A_0 = E\sigma$ and $t\sigma \in A_k$.

By Proposition 14 and Lemma 14 (which also works for DY case) we can easily prove that if $k > 0$, $\delta(A_j) \subseteq \mathrm{Der}_{DY}(\delta(A_{j-1}))$, $j = 1, \ldots, k$. Note, that $\delta(A)$ is a set of standard terms (and thus, normalized) for any set of terms A. Then, applying the transitivity property of $\mathrm{Der}_{DY}(\cdot)$ (Lemma 14 for DY) k times, we have that $\delta(A_k) \subseteq \mathrm{Der}_{DY}(\delta(A_0))$. In the case where $k = 0$, the statement $\delta(A_k) \subseteq \mathrm{Der}_{DY}(\delta(A_0))$ is also true.

Using Proposition 15 we have that $\delta(A_0) = \delta(E\sigma) = E\delta(\sigma)$, as $E \subseteq \mathrm{QSub}(\mathcal{S})$. The same for t: $\delta(t\sigma) = t\delta(\sigma)$, and as $t\sigma \in A_k$, we have $t\delta(\sigma) \in \delta(A_k)$.

Thus, we have that $t\delta(\sigma) \in \delta(A_k) \subseteq \mathrm{Der}_{DY}(\delta(A_0)) = \mathrm{Der}_{DY}(E\delta(\sigma))$, that means $\delta(\sigma)$ DY-satisfies any constraint of $\mathcal{S}$. □

We present an example illustrating the theorem.

Example 8. Let us consider a standard constraint system similar to Example 5.

$$\mathcal{S} = \left\{ \begin{array}{ll} \mathrm{enc}(x,a), \mathrm{pair}(c,a) & \rhd\ b \\ \mathrm{pair}(x,c) & \rhd\ a \end{array} \right\},$$

Using Algorithm 1, we can get a model of $\mathcal{S}$ within DY+ACI, as in Example 6, $\sigma = \{x \mapsto \cdot(a,b,c)\}$. Then, by applying the transformation $\delta(\cdot)$, we will get $\sigma' = \delta(\sigma) = \{x \mapsto \mathrm{pair}(a, \mathrm{pair}(b,c))\}$. We can see that σ' is also a model of $\mathcal{S}$ within DY (as it was proven in Theorem 2).

Corollary 16 (of Theorem 2 and Proposition 13)**.** *A standard constraint system $\mathcal{S}$ is satisfiable within DY iff it is satisfiable within DY+ACI.*

Theorem 3. Satisfiability of DY constraint systems is in *NP*.

Proof. It is a consequence of Proposition 12 and Corollary 16.
□

Then, since the problem is known to be *NP*-hard Rusinowitch and Turuani (2003) for the particular case of well-formed constraints, we obtain the *NP*-hardness and thus, *NP*-completeness:

Corollary 17. *Satisfiability of DY constraint systems is NP-complete.*

6. Conclusions

In this work we have presented a decision algorithm for the satisfiability of general constraint systems within Dolev-Yao deduction system possibly extended with an ACI symbol that can be used to represent sets of terms. The complexity of the algorithm was proved to be in *NP*.

We have also given two applications of the presented result: protocol insecurity with non-communicating intruders and discovering XML-based attacks. Moreover, recent works Mödersheim et al. (2013); Kassem et al. (2013) have shown the interest of general intruder constraints for analyzing security problems of mobile code encountered in web-browsers, smartphones, and virtualized infrastructures. In this approach, an attacker can construct new processes from his knowledge by using the constructors of the ambient calculus, in the same way as a Dolev Yao intruder constructs messages in protocol verification.

As future works one may consider the verification of equivalence properties such as anonymity or secrecy of a ballot in vote. For instance one has to show that there is no action for an attacker that makes distinguishable two protocol executions with different identities or vote values. In presence of multiple intruders this may require to extend a decision procedure for trace equivalences Baudet (2005) using our techniques. Another challenging extension of our result would be a decision procedure for general constraints with negation. Negation constraints are particularly useful to model non-disclosure policies when generating distributed orchestration of secure services, following the approach introduced in Avanesov et al. (2012).

A. General constraints for subterm theories

We demonstrate here that the well-formedness property is a strong restriction for constraint systems: in some theories satisfiability of well-formed constraints may be decidable, while satisfiability of general constraints may be undecidable.

We define a *subterm deduction system* to be a finite set of rules of the following forms:
(1) composition rules: for all *public* functional symbols f, $x_1, \ldots, x_k \to f(x_1, \ldots, x_k)$
(2) decomposition rules: $t_1, \ldots, t_m \to s$, where s is a subterm of t_i for some i.

We show that the satisfiability of a constraint system within a subterm deduction system is undecidable in general. More precisely:
Instance: a subterm deduction system D, a constraint system C.
Question: is C satisfiable?
To show this, we reduce the halting problem of a Deterministic Turing Machine (TM) M that works on a single tape. We consider the tape alphabet $\Gamma = \{0, 1, \flat\}$, and $\flat$ is the

blank symbol. The states of the TM M are in a finite set $Q = \{q_1, q_2, \ldots, q_n\}$. W.l.o.g. we can assume that q_1 (resp. q_n) is the unique initial (resp. accepting) state.

In order to represent Turing machine configurations as terms we shall introduce a set of variables $\mathcal{X}$ and an alphabet $\mathcal{F}$

$$\mathcal{F} := \{0, 1, \flat, \bot\} \cup Q,$$

where $\mathcal{F} \setminus \{\bot\}$ are public functional symbols.

The TM configuration with tape $\bot$ *abcde* $\bot$, (where $\bot$ is an endmarker), with symbol d under the head, and state q will be represented by the following term of $q(c(b(a(\bot), d(e(\bot), x)$ where $x \in \mathcal{X}$ and $a, b, c, d, e \in \{0, 1, \flat\}$.

The composition rules we consider for the TM are $u \to f(u)$ for each $f \in \{0, 1, \flat\}$ and $u, v, w \to q(u, v, w)$ for each $q \in Q$. For each TM transition of M we will introduce some decomposition deduction rule that can be applied on a term representation $q(u, v, q'(u', v', x'))$ iff the transition can be applied to a configuration represented by $q(u, v, \_)$ and generate a configuration represented by $q'(u', v', \_)$.

For each TM instruction of type: *"In state q reading a go to state q′ and write b"*, we define the following rule for $a, b \in \{0, 1, \flat\}$:

$$q(u, a(v), q'(u, b(v), x)) \to q'(u, b(v), x).$$

For each instruction of type: *"In state q reading a go to state q′ and move right"*, we define the following rules for $a \in \{0, 1, \flat\}$:

$$q(u, a(v), q'(a(u), v, x)) \to q'(a(u), v, x).$$

A rule is for extending the tape on the right when needed:

$$q(u, \bot, q'(u, \flat(\bot), x)) \to q'(u, \flat(\bot), x).$$

For each instruction of type: *"In state q reading a go to state q′ and move left"*, we define the following rules for $a \in \{0, 1, \flat\}$ for each $b \in \{0, 1, \flat\}$:

$$q(b(u), a(v), q'(u, b(a(v)), x)) \to q'(u, b(a(v)), x).$$

A rule is for extending the tape on the left when needed:

$$q(\bot, v, q'(\bot, \flat(v), x)) \to q'(\bot, \flat(v), x).$$

The resulting deduction system D_M is obviously a subterm deduction system.

Let us consider a constraint $\mathcal{S}$ to be solved modulo D_M:

$$\{q_1(\bot, \bot, x)\} \rhd q_n(\bot, \bot, y).$$

This constraint is satisfiable iff there is a sequence of transitions of M from a configuration with initial state q_1 and empty tape to a configuration with an accepting state and empty tape. Hence the constraint solving problem is undecidable.

Let us recall the definition of some properties of constraint systems. These two properties are natural for modeling standard security protocols:

Variable origination: $\forall i,\ \forall x \in \mathrm{Vars}\,(E_i)\ \ \exists j < i\ \ x \in \mathrm{Vars}\,(t_j)$,

Monotonicity: $j < i \implies E_j \subseteq E_i$.

Note that $\{\{q_1(\bot,\bot,x)\} \rhd q_n(\bot,\bot,y)\}$ is obviously monotonic.

As a consequence, satisfiability of monotonic constraint systems (but without variable origination) is undecidable. Here is another constraint system, where variable origination is satisfied, but monotony is not. It can be used for reducing the halting problem again:

$$\{\{\bot\} \rhd x,\ \{q_1(\bot,\bot,x)\} \rhd q_n(\bot,\bot,y)\}$$

As a consequence, satisfiability of constraint systems with variable origination (but without monotonicity) is undecidable.

We should note by contrast (see Baudet (2005)) that constraint solving in subterm convergent theories is decidable if the constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ satisfies both variable origination and monotonicity. Moreover, the problem of constraint system satisfiability for a subterm deduction system encoding a Deterministic Turing Machine as above can be reduced to the problem of constraint system satisfiability within a subterm convergent equational theory.

If we consider a Universal Turing Machine [4] with a single final state q_n and that halts with an empty tape if and only if the Turing Machine with the given code terminates on the given input, then by reusing the encoding above we will obtain a fixed subterm deduction system D_U for which the following problem is undecidable:
Instance: A constraint system C.
Question: is C satisfiable within D_U?

This is due to the fact that the halting problem of a TM with code $TMcode$ on input $Input$ is encoded into solving the following constraint:

$$\{q_1(\bot, TMcode + Input, x)\} \rhd q_n(\bot,\bot,y).$$

B. Proofs

B.1. Proof of Lemma 12

By induction on $|\mathrm{Sub}\,(t)|$.

- $|\mathrm{Sub}\,(t)| = 1$, implies $t = a \in \mathcal{A}$ and then $\mathrm{elems}\,(a) = \{a\}$, i.e. the equality becomes trivial.
- Suppose that for any $t : |\mathrm{Sub}\,(t)| < k$ $(k > 1)$, $H\,(t) = \bigcup_{p\in\mathrm{elems}(t)} H\,(p)$ holds.
- Given a term $t : |\mathrm{Sub}\,(t)| = k$, $k > 1$. We should prove $H\,(t) = \bigcup_{p\in\mathrm{elems}(t)} H\,(p)$.
  - $t = \mathrm{priv}\,(t_1)$ or $t = \mathrm{bin}\,(p,q)$. In both cases, $\mathrm{elems}\,(t) = \{t\}$, and thus, he equality is trivial.
  - $t = \cdot(L)$. Note that $\forall s \in L$, $|\mathrm{Sub}\,(s)| < k$. Then, on one hand, $H\,(\cdot\,(L)) = \bigcup_{p\in L} H\,(p) =$ (by induction hypothesis) $= \bigcup_{p\in L}\bigcup_{p'\in\mathrm{elems}(p)} H\,(p')$. On the other hand, $\bigcup_{p\in\mathrm{elems}(\cdot(L))} H\,(p) = \bigcup_{p\in\bigcup_{p'\in L}\mathrm{elems}(p')} H\,(p) = \bigcup_{p'\in L}\bigcup_{p\in\mathrm{elems}(p')} H\,(p)$. Thus, $H\,(t) = \bigcup_{p\in\mathrm{elems}(t)} H\,(p)$.

[4] Thanks to R. Küsters for the idea.

B.2. Proof of Lemma 6

First we prove that $\mathrm{elems}(t) \subseteq \mathrm{QSub}(t)$. We use a proof by induction on $|\mathrm{Sub}(t)|$.

- If $\mathrm{root}(t) \neq \cdot$, then $\mathrm{elems}(t) = \{t\} \subseteq \mathrm{QSub}(t)$. This case includes all t such that $|\mathrm{Sub}(t)| = 1$. Thus we need to consider only $t = \cdot(L)$.
- Suppose that for any $t : |\mathrm{Sub}(t)| < k$ $(k \geq 1)$, the statement holds.
- If for some t we have $|\mathrm{Sub}(t)| = k$, $k > 1$, then $\mathrm{elems}(t) = \bigcup_{p \in L} \mathrm{elems}(p)$ and $\mathrm{QSub}(t) = \{t\} \bigcup_{p \in L} \mathrm{QSub}(p)$. And since $|\mathrm{Sub}(p)| < k$ using the induction supposition we obtain the wanted statement.

Now we show that $\mathrm{QSub}(t) \subseteq \mathrm{Sub}(t)$. Again, applying proof by induction on $|\mathrm{Sub}(t)|$ we have:

- If $|\mathrm{Sub}(t)| = 1$, then $\mathrm{QSub}(t) = \mathrm{Sub}(t) = \{t\}$.
- Suppose that for any $t : |\mathrm{Sub}(t)| < k$ $(k \geq 1)$, the statement holds.
- If for some t we have $|\mathrm{Sub}(t)| = k$, $k > 1$, then
  - $t = \mathrm{bin}(t_1, t_2)$.Then $\mathrm{QSub}(t) = \{t\} \cup \mathrm{QSub}(t_1) \cup \mathrm{QSub}(t_2)$ and $\mathrm{Sub}(t) = \{t\} \cup \mathrm{Sub}(t_1) \cup \mathrm{Sub}(t_2)$, where $\max\{|\mathrm{Sub}(t_1)|, |\mathrm{Sub}(t_2)|\} < k$. And then using induction supposition we can conclude for this case.
  - $t = \mathrm{priv}(t_1)$. Proof is similar to one for the case above.
  - $t = \cdot(t_1, \ldots, t_m)$. Then we have $\mathrm{QSub}(t) = \{t\} \cup \bigcup_{p \in \mathrm{elems}(t_1, \ldots, t_m)} \mathrm{QSub}(p) \subseteq$ (using the already proved part of the property) $\subseteq \{t\} \cup \bigcup_{p \in \mathrm{QSub}(\{t_1, \ldots, t_m\})} \mathrm{QSub}(p) =$ (as $\mathrm{QSub}(\mathrm{QSub}(t)) = \mathrm{QSub}(t)$) $= \{t\} \cup \bigcup_{p \in \{t_1, \ldots, t_m\}} \mathrm{QSub}(p) \subseteq$ (using induction supposition, as $\forall i\, |\mathrm{Sub}(t_i)| < k$) $\subseteq \{t\} \cup \bigcup_{p \in \{t_1, \ldots, t_m\}} \mathrm{Sub}(p) = \mathrm{Sub}(t)$.

B.3. Proof of Proposition 12

As was stated before, the measure of the problem input is $\mathrm{measure}(\mathcal{S}) = n \times |\mathrm{Sub}(\mathcal{S})| + |\mathbb{E}(\mathcal{S})|$, where $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$.

Algorithm 1 returns a normalized proof σ for the decision problem if it exists. Moreover, $|\mathrm{Sub}(x\sigma)| \leq 2 \times |\mathrm{Sub}(\mathcal{S})|$ for any $x \in \mathrm{Vars}(\mathcal{S})$.

First, we will normalize $\mathcal{S}\sigma$. From Lemma 21 follows, that we can do it for the time $T_{\ulcorner\urcorner} \leq P_{\ulcorner\urcorner}(\mathrm{measure}(\mathcal{S}\sigma))$, where $P_{\ulcorner\urcorner}$ is some polynomial with non-negative coefficients of some degree $m'' > 0$.

From § 3.3 we know that checking the derivability of a normalized ground term g from a set of normalized ground terms G takes a polynomial time (depending on $|\mathrm{Sub}(G \cup \{g\})|$). That is, there exists a polynomial P_g with non-negative coefficients, such that the number of operations (execution time) to verify the derivability (g from G) will be limited by $P_g(|\mathrm{Sub}(G \cup \{g\})|)$. Then the execution time for checking a set of ground constraints $\{G_i \rhd g_i\}_{i=1,\ldots,n}$ will be limited by $\sum_{i=1}^{n} P_g(|\mathrm{Sub}(G_i \cup \{g_i\})|)$.

To prove that the algorithm is in NP we need to show that the execution time for checking a solution is polynomially limited by the input measure, i.e. there exists a polynomial P, such that the execution time does not exceed $O(P(n \times |\mathrm{Sub}(\mathcal{S})| + |\mathbb{E}(\mathcal{S})|))$ steps.

In our case, the execution time T of a check will be $T = T_{\ulcorner\urcorner} + T_{val} + T_g$, where T_{val} is the time needed to check the validity of terms in $\mathcal{S}\sigma$, and T_g is the time needed for checking ground derivability of $\mathcal{S}\sigma$: $T_g \leq \sum_{i=1}^{n} P_g(|\mathrm{Sub}(\ulcorner(E_i \cup \{t_i\})\sigma\urcorner)|)$. As P_g is a polynomial, let us say, of degree $m' > 0$, with non-negative coefficients, we can use the fact that for any positive integers $x_1, \ldots, x_k$ we have $\sum_{i=1}^{k} P_g(x_i) \leq P_g(\sum_{i=1}^{k} x_i)$.

Then we have $T_g \leq P_g(\sum_{i=1}^{n}|\mathrm{Sub}\,(\ulcorner(E_i \cup \{t_i\})\sigma\urcorner)|)$ and by Lemma 22 we have $T_g \leq P_g(\sum_{i=1}^{n}|\mathrm{Sub}\,((E_i \cup \{t_i\})\sigma)|)$; using the same lemma, we have

$$\begin{aligned}
T_g &\leq P_g\left(\sum_{i=1}^{n}\left(|\mathrm{Sub}\,(E_i \cup \{t_i\})| + |\mathrm{Sub}\left(\bigcup_x x\sigma\right)|\right)\right) \leq \\
&\leq P_g\left(\sum_{i=1}^{n}\left(|\mathrm{Sub}\,(E_i)| + |\mathrm{Sub}\,(t_i)| + \sum_x |\mathrm{Sub}\,(x\sigma)|\right)\right) \leq \\
&\leq P_g\left(\sum_{i=1}^{n}(2 \times |\mathrm{Sub}\,(\mathcal{S})|) + n \times \sum_x (|\mathrm{Sub}\,(x\sigma)|)\right) \leq \\
&\leq P_g\left(2 \times n \times |\mathrm{Sub}\,(\mathcal{S})| + n \times \sum_x (2 \times |\mathrm{Sub}\,(\mathcal{S})|)\right) \leq \\
&\leq P_g\left(2 \times n \times |\mathrm{Sub}\,(\mathcal{S})| + 2 \times n \times (|\mathrm{Sub}\,(\mathcal{S})|)^2\right) \leq \\
&\leq P_g\left(4 \times n \times (|\mathrm{Sub}\,(\mathcal{S})|)^2\right) \leq P_g\left(4 \times (n \times |\mathrm{Sub}\,(\mathcal{S})| + |\mathbb{E}\,(\mathcal{S})|)^2\right) = \\
&= O\left((\mathrm{measure}\,(\mathcal{S}))^{2m'}\right).
\end{aligned}$$

On the other hand, let us consider $T_{\ulcorner\urcorner}$. We have $T_{\ulcorner\urcorner} \leq P_{\ulcorner\urcorner}(n \times |\mathrm{Sub}\,(\mathcal{S}\sigma)| + |\mathbb{E}\,(\mathcal{S}\sigma)|)$. One can see that the number of edges in DAG-representation of $\mathcal{S}\sigma$ (where every variable x of $\mathcal{S}$ is replaced by $x\sigma$) will not exceed the number of edges in $\mathcal{S}$ plus the number of edges of all $x\sigma$: $|\mathbb{E}\,(\mathcal{S}\sigma)| \leq |\mathbb{E}\,(\mathcal{S})| + \sum_{x\in\mathrm{Vars}(\mathcal{S})}|\mathbb{E}\,(x\sigma)|$. And since σ is normalized, we can use Lemma 20: $T_{\ulcorner\urcorner} \leq P_{\ulcorner\urcorner}(n \times |\mathrm{Sub}\,(\mathcal{S}\sigma)| + |\mathbb{E}\,(\mathcal{S})| + \sum_{x\in\mathrm{Vars}(\mathcal{S})}(|\mathrm{Sub}\,(x\sigma)|)^2)$.

Then, using Lemma 5 we obtain $\mathrm{Sub}\,(\mathcal{S}\sigma) = \mathrm{Sub}\,(\mathcal{S})\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(\mathcal{S})\,\sigma)$, and thus, $|\mathrm{Sub}\,(\mathcal{S}\sigma)| \leq |\mathrm{Sub}\,(\mathcal{S})\,\sigma| + \sum_{x\in\mathrm{Vars}(\mathcal{S})}|\mathrm{Sub}\,(x\sigma)|$. From $\forall T \subseteq \mathcal{T}, |T\sigma| \leq |T|$ follows $|\mathrm{Sub}\,(\mathcal{S})\,\sigma| \leq |\mathrm{Sub}\,(\mathcal{S})|$. Since $|\mathrm{Sub}\,(x\sigma)| \leq 2\times|\mathrm{Sub}\,(\mathcal{S})|$ and $|\mathrm{Vars}\,(\mathcal{S})| \leq |\mathrm{Sub}\,(\mathcal{S})|$, we obtain $|\mathrm{Sub}\,(\mathcal{S}\sigma)| \leq |\mathrm{Sub}\,(\mathcal{S})| + 2\times(|\mathrm{Sub}\,(\mathcal{S})|)^2$. In the same way, $\sum_{x\in\mathrm{Vars}(\mathcal{S})}(|\mathrm{Sub}\,(x\sigma)|)^2 \leq |\mathrm{Sub}\,(\mathcal{S})| \times (2 \times |\mathrm{Sub}\,(\mathcal{S})|)^2$. Therefore, $T_{\ulcorner\urcorner} \leq P_{\ulcorner\urcorner}(n \times (|\mathrm{Sub}\,(\mathcal{S})| + 2 \times (|\mathrm{Sub}\,(\mathcal{S})|)^2) + |\mathbb{E}\,(\mathcal{S})| + |\mathrm{Sub}\,(\mathcal{S})| \times (2 \times |\mathrm{Sub}\,(\mathcal{S})|)^2) = O\left((\mathrm{measure}\,(\mathcal{S}))^{3m''}\right)$.

Note that once $\mathcal{S}\sigma$ is normalized, the validity can be checked in linear time $T_{val} = O(\mathrm{measure}\,(\mathcal{S}\sigma))$. Using the same reasoning as for $T_{\ulcorner\urcorner}$, we obtain $T_{val} = O(\mathrm{measure}\,(\mathcal{S})^3)$.

Summing up, $T = O\left((\mathrm{measure}\,(\mathcal{S}))^{3m''} + (\mathrm{measure}\,(\mathcal{S}))^{2m'} + (\mathrm{measure}\,(\mathcal{S}))^3\right)$. This shows that checking a solution returned by the algorithm takes polynomial time, giving the expected complexity.

B.4. Proof of Proposition 3

Let us consider all the cases of DY+ACI rules:

- $t_1, t_2 \to \ulcorner\mathrm{pair}\,(t_1, t_2)\urcorner$ We have two cases:
  - $\exists u \in \mathring{\mathrm{Sub}}\,(S)$ such that $\ulcorner\mathrm{pair}\,(t_1, t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then we have $\pi(H\,(\ulcorner\mathrm{pair}\,(t_1, t_2)\urcorner)) = \pi(H\,(\mathrm{pair}\,(\ulcorner t_1\urcorner, \ulcorner t_2\urcorner))) = \pi(\{\mathrm{pair}\,(\pi(H\,(\ulcorner t_1\urcorner)), \pi(H\,(\ulcorner t_2\urcorner)))\}) = \ulcorner\mathrm{pair}\,(\pi(H\,(t_1)), \pi(H\,(t_2)))\urcorner$ and then $\pi(H\,(\ulcorner\mathrm{pair}\,(t_1, t_2)\urcorner)) \in \mathrm{Der}\,(\{\pi(H\,(t_1)), \pi(H\,(t_2))\})$.
  - $\nexists u \in \mathring{\mathrm{Sub}}\,(S)$ such that $\ulcorner\mathrm{pair}\,(t_1, t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then (by definition, Lemma 13 and Lemma 10) $\pi(H\,(\ulcorner\mathrm{pair}\,(t_1, t_2)\urcorner)) = \pi(H\,(\ulcorner t_1\urcorner) \cup H\,(\ulcorner t_2\urcorner)) \in \mathrm{Der}\,(\ulcorner H\,(t_1) \cup H\,(t_2)\urcorner)$. By Lemma 10, $\ulcorner H\,(t_1)\urcorner \subseteq \mathrm{Der}\,(\{\pi(H\,(t_1))\})$ and $\ulcorner H\,(t_2)\urcorner \subseteq \mathrm{Der}\,(\{\pi(H\,(t_2))\})$, then

by Lemma 14, $\ulcorner H(t_1)\urcorner \cup \ulcorner H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(t_1))\} \cup \{\pi(H(t_2))\})$. Now, by applying Lemma 14, we have $\pi(H(\ulcorner \mathrm{pair}(t_1,t_2)\urcorner)) \in \mathrm{Der}(\{\pi(H(t_1))\} \cup \{\pi(H(t_2))\})$. So, in this case $\pi(H(r)) \in \mathrm{Der}(\{\pi(H(l_1)), \pi(H(l_2))\})$.

- $t_1, t_2 \to \ulcorner \mathrm{enc}(t_1,t_2)\urcorner$. Proof of this case can be done by analogy of previous one.
- $\{t_1,t_2\} \to \ulcorner \mathrm{aenc}(t_1,t_2)\urcorner$. The same.
- $t_1, \mathrm{priv}(t_2) \to \ulcorner \mathrm{sig}(t_1, \mathrm{priv}(t_2))\urcorner$.
  - $\exists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{sig}(t_1,\mathrm{priv}(t_2))\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\ulcorner \mathrm{sig}(t_1,\mathrm{priv}(t_2))\urcorner)) = \pi(H(\mathrm{sig}(t_1,\mathrm{priv}(t_2)))) = \pi(\{\mathrm{sig}(\pi(H(\ulcorner t_1\urcorner)), \pi(H(\ulcorner \mathrm{priv}(t_2)\urcorner)))\}) = \ulcorner \mathrm{sig}(\pi(H(t_1)), \mathrm{priv}(\pi(H(t_2))))\urcorner$ and then $\pi(H(\ulcorner \mathrm{sig}(t_1,\mathrm{priv}(t_2))\urcorner)) \in \mathrm{Der}(\{\pi(H(t_1)), \pi(H(\mathrm{priv}(t_2)))\})$ (as $\pi(H(\mathrm{priv}(t_2))) = \mathrm{priv}(\pi(H(t_2)))$).
  - $\nexists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{sig}(t_1,\mathrm{priv}(t_2))\urcorner = \ulcorner u\sigma\urcorner$. This case can be proved in similar way as done for $\{t_1,t_2\} \to \ulcorner \mathrm{pair}(t_1,t_2)\urcorner$.
- $t_1, \ldots, t_m \to \ulcorner \cdot(t_1,\ldots,t_m)\urcorner$. On one hand, $\pi(H(\ulcorner \cdot(t_1,\ldots,t_m)\urcorner)) = \pi(H(\cdot(t_1,\ldots,t_m))) = \pi(H(t_1) \cup \cdots \cup H(t_m)) \in \mathrm{Der}(\ulcorner H(t_1) \cup \cdots \cup H(t_m)\urcorner)$. On the other hand, $\ulcorner H(t_i)\urcorner \subseteq \mathrm{Der}(\{\pi(H(t_i))\})$. And thus, by Lemma 14, $\pi(H(\ulcorner \cdot(t_1,\ldots,t_m)\urcorner)) \in \mathrm{Der}(\{\pi(H(t_1)), \ldots, \pi(H(t_m))\})$.
- $\mathrm{enc}(t_1,t_2), \ulcorner t_2\urcorner \to \ulcorner t_1\urcorner$. Here we have to show that $\pi(H(\ulcorner t_1\urcorner))$ is derivable from $\{\pi(H(\mathrm{enc}(t_1,t_2))), \pi(H(\ulcorner t_2\urcorner))\}$. Consider two cases:
  - $\exists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{enc}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{enc}(t_1,t_2))) = \mathrm{enc}(\pi(H(t_1)), \pi(H(t_2)))$, and $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\{\mathrm{enc}(\pi(H(t_1)), \pi(H(t_2))), \ulcorner \pi(H(\ulcorner t_2\urcorner))\urcorner\})$.
  - $\nexists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{enc}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{enc}(t_1,t_2))) = \pi(H(t_1) \cup H(t_2))$. From Lemma 10, we have $\ulcorner H(t_1) \cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{enc}(t_1,t_2)))\})$, thus $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{enc}(t_1,t_2)))\})$. And then, by Lemma 10 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 13 and Lemma 14, we have $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\pi(H(\mathrm{enc}(t_1,t_2))))$.
- $\mathrm{aenc}(t_1,t_2), \ulcorner \mathrm{priv}(t_2)\urcorner \to \ulcorner t_1\urcorner$. Here we have to show that $\pi(H(\ulcorner t_1\urcorner))$ is derivable from $\{\pi(H(\mathrm{aenc}(t_1,t_2))), \pi(H(\ulcorner \mathrm{priv}(t_2)\urcorner))\}$. Consider two cases:
  - $\exists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{aenc}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{aenc}(t_1,t_2))) = \mathrm{aenc}(\pi(H(t_1)), \pi(H(t_2)))$, and then $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\{\mathrm{aenc}(\pi(H(t_1)), \pi(H(t_2))), \ulcorner \mathrm{priv}(\pi(H(t_2)))\urcorner\})$. Meanwhile, $\pi(H(\ulcorner \mathrm{priv}(t_2)\urcorner)) = \pi(H(\mathrm{priv}(t_2))) = \pi(\{\mathrm{priv}(\pi(H(t_2)))\}) = \ulcorner \mathrm{priv}(\pi(H(t_2)))\urcorner$.
  - $\nexists u \in \mathrm{QSub}(S)$ such that $\ulcorner \mathrm{aenc}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{aenc}(t_1,t_2))) = \pi(H(t_1) \cup H(t_2))$. By Lemma 10 we have $\ulcorner H(t_1) \cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{aenc}(t_1,t_2)))\})$, thus $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{aenc}(t_1,t_2)))\})$. And then, by Lemma 10 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 13 and Lemma 14, $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\pi(H(\mathrm{aenc}(t_1,t_2))))$.
- $\mathrm{pair}(t_1,t_2) \to \ulcorner t_1\urcorner$. Here, as usual, we consider two cases:
  - $\exists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{pair}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{pair}(t_1,t_2))) = \mathrm{pair}(\pi(H(t_1)), \pi(H(t_2)))$ and then $\pi(H(\ulcorner t_1\urcorner)) = \ulcorner \pi(H(t_1))\urcorner \in \mathrm{Der}(\{\pi(H(\mathrm{pair}(t_1,t_2)))\})$.
  - $\nexists u \in \overset{\circ}{\mathrm{Sub}}(S)$ such that $\ulcorner \mathrm{pair}(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{pair}(t_1,t_2))) = \pi(H(t_1) \cup H(t_2))$. Then by Lemma 10 we obtain $\ulcorner H(t_1) \cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{pair}(t_1,t_2)))\})$, thus $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{pair}(t_1,t_2)))\})$. And then, by Lemma 10 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 14, $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\pi(H(\mathrm{pair}(t_1,t_2))))$.

- $\mathrm{pair}\,(t_1, t_2) \to \ulcorner t_2 \urcorner$. Proof like above.
- $\cdot\,(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$. We have $\pi(H\,(\cdot\,(t_1, \ldots, t_2))) = \pi(H\,(t_1) \cup \cdots \cup H\,(t_m))$. Then by Lemma 10, $\ulcorner H\,(t_1) \cup \cdots \cup H\,(t_m) \urcorner \subseteq \mathrm{Der}\,(\pi(H\,(\cdot\,(t_1, \ldots, t_m))))$; thus $\ulcorner H\,(t_i) \urcorner \subseteq \mathrm{Der}\,(\pi(H\,(\cdot\,(t_1, \ldots, t_m))))$. As $\pi(H\,(t_i)) \in \mathrm{Der}\,(\ulcorner H\,(t_i) \urcorner)$, by Lemma 14 we have $\pi(H\,(\ulcorner t_i \urcorner)) = \pi(H\,(t_i)) \in \mathrm{Der}\,(\pi(H\,(\cdot\,(t_1, \ldots, t_2))))$.

As all possible cases satisfy lemma conditions, we proved the lemma.

B.5. Proof of Lemma 22

First, we show an auxiliary statement:

Lemma 25. For any term t, $\forall s \in \mathrm{Sub}\,(\ulcorner t \urcorner)\, \exists s' \in \mathrm{Sub}\,(t) \,:\, s = \ulcorner s' \urcorner$

Proof. Suppose the opposite and let us take $s \in \mathrm{Sub}\,(\ulcorner t \urcorner)$ with maximal $|\mathrm{Sub}\,(s)|$ that does not satisfy the desired property. Note that the "biggest" term in $\mathrm{Sub}\,(\ulcorner t \urcorner)$, i.e. $\ulcorner t \urcorner$, does satisfy the property, as we can choose $s' = t \in \mathrm{Sub}\,(t)$. By definition of $\mathrm{Sub}\,(\cdot)$ if $s \in \mathrm{Sub}\,(\ulcorner t \urcorner)$ and $s \neq \ulcorner t \urcorner$ then $\exists r \in \mathrm{Sub}\,(\ulcorner t \urcorner)$ such that

- $r = \mathrm{bin}\,(p, s)$ or $r = \mathrm{bin}\,(s, p)$ or $r = \mathrm{priv}\,(s)$. Without loss of generality we consider only the first case ($r = \mathrm{bin}\,(p, s)$) as other ones are similar. As $|\mathrm{Sub}\,(r)| > |\mathrm{Sub}\,(s)|$, there exists $r' \in \mathrm{Sub}\,(t)$ such that $r = \ulcorner r' \urcorner$. By definition of $\ulcorner \cdot \urcorner$:
  - either $r' = \mathrm{bin}\,(p', s')$ and $\ulcorner p' \urcorner = p$ and $\ulcorner s' \urcorner = s$. As $s' \in \mathrm{Sub}\,(r') \subseteq \mathrm{Sub}\,(t)$ the property is proved.
  - or $r' = \cdot\,(L)$ and $\ulcorner \mathrm{elems}\,(L) \urcorner = \{r\}$. Since $\forall q \in \mathrm{elems}\,(L)\,,\ \mathrm{root}\,(q) \neq \cdot$, then $\exists q \in \mathrm{elems}\,(L) : q = \mathrm{bin}\,(p', s')$ and $\ulcorner p' \urcorner = p$ and $\ulcorner s' \urcorner = s$. Using Lemma 6 we have $s' \in \mathrm{Sub}\,(t)$.
- $r = \cdot\,(L)$ and $s \in L$. Then (since $|\mathrm{Sub}\,(r)| > |\mathrm{Sub}\,(s)|$) we have $\exists r' \in \mathrm{Sub}\,(t) : \ulcorner r' \urcorner = r$. Thus r is normalized ($r = \ulcorner r' \urcorner = \ulcorner \ulcorner r' \urcorner \urcorner = \ulcorner r \urcorner$), and thus, $\mathrm{root}\,(s) \neq \cdot$. Then by definition of $\ulcorner \cdot \urcorner$ we have $r' = \cdot\,(L')$ and $L \approx \ulcorner \mathrm{elems}\,(L') \urcorner$, and thus, $s \in \ulcorner \mathrm{elems}\,(L') \urcorner$, that is $\exists s' \in \mathrm{elems}\,(L') : s = \ulcorner s' \urcorner$. Using again Lemma 6 we have $s' \in \mathrm{Sub}\,(t)$.

□

Then, using Lemma 25 and the fact that $\ulcorner \cdot \urcorner$ is deterministic, we obtain $\forall p, q \in \mathrm{Sub}\,(\ulcorner t \urcorner)\, p \neq q\, \exists p', q' \in \mathrm{Sub}\,(t) : p = \ulcorner p' \urcorner \wedge q = \ulcorner q' \urcorner \wedge p \neq q$. And thus, $|\mathrm{Sub}\,(\ulcorner t \urcorner)| \leq |\mathrm{Sub}\,(t)|$.

B.6. Proof of Proposition 8

Proof. From Proposition 1 we know that if σ' is a model of $\mathcal{S}$ then $\ulcorner \sigma' \urcorner$ is a model of $\mathcal{S}$ and $\ulcorner \sigma' \urcorner$ is a model of $\ulcorner \mathcal{S} \urcorner$. Then, there exists a substitution $\theta : \mathrm{dom}\,(\theta) = \mathrm{dom}\,(\ulcorner \sigma' \urcorner)\,, \mathrm{img}\,(\theta) \subseteq \mathrm{dom}\,(\theta)\,, \sigma'' = \ulcorner \sigma' \urcorner|_{\mathrm{img}(\theta)}$ (where $\theta|_V$ is a substitution obtained from θ by narrowing its domain to set V) and σ'' is a model of $\ulcorner \mathcal{S} \urcorner \theta$ such that $x\sigma'' \neq y\sigma''$, if $x \neq y$ (this is true because we can show how to build θ : given the $\ulcorner \sigma' \urcorner$ — simply split $\mathrm{dom}\,(\ulcorner \sigma' \urcorner)$ into the classes of equivalence modulo $\ulcorner \sigma' \urcorner$, i.e. $x \equiv y \iff x \ulcorner \sigma' \urcorner = y \ulcorner \sigma' \urcorner$; for every class choose one representative $[x]_\equiv$, and then $x\theta = [x]_\equiv$). Note that $\theta\sigma'' = \sigma'$, that's why σ'' is a model of $\ulcorner \mathcal{S} \urcorner \theta$.

Then, as σ'' is a model of $\ulcorner \mathcal{S} \urcorner \theta$, using Proposition 1, we can say that σ'' is a model of $\ulcorner \ulcorner \mathcal{S} \urcorner \theta \urcorner$. Moreover, $x\sigma'' \neq y\sigma''$, if $\forall x, y \in \mathrm{dom}\,(\sigma'')\ x \neq y$ and σ'' is normalized. Then, we can apply Corollary 6, which gives us existence of conservative model δ of $\ulcorner \ulcorner \mathcal{S} \urcorner \theta \urcorner$. That is why we can apply Proposition 7: $\forall x \in \mathrm{Vars}\,(\ulcorner \ulcorner \mathcal{S} \urcorner \theta \urcorner)\,, |\mathrm{Sub}\,(x\delta)| \leq 2 \times |\mathrm{Sub}\,(\ulcorner \ulcorner \mathcal{S} \urcorner \theta \urcorner)|$.

Note that using Proposition 1, Lemma 19 and definition of "model", we can easily show that $\delta[\theta]$ is a model of $\ulcorner S \urcorner$. Moreover, $\delta[\theta]$ is normalized. By definition of $\delta[\theta]$ we can say that $\forall x \in \mathrm{dom}\,(\delta[\theta])\ \exists y \in \mathrm{dom}\,(\theta)\, \theta : x\delta[\theta] = y\delta$ and as $y \in \mathcal{X}$ (by definition of θ), then $|\mathrm{Sub}\,(x\delta[\theta])| = |\mathrm{Sub}\,(y\delta)| \leq 2 \times |\mathrm{Sub}\,(\ulcorner\ulcorner \mathcal{S} \urcorner \theta \urcorner)| \leq 2 \times |\mathrm{Sub}\,(\ulcorner \mathcal{S} \urcorner \theta)|$. Applying Lemma 18, we have $|\mathrm{Sub}\,(x\delta[\theta])| \leq 2 \times |\mathrm{Sub}\,(\ulcorner \mathcal{S} \urcorner)|$.

Summing up, we have a normalized model $\sigma = \delta[\theta]$ of $\ulcorner \mathcal{S} \urcorner$ such that for any $x \in \mathrm{dom}\,(\sigma)$ we have $|\mathrm{Sub}\,(x\sigma)| \leq 2 \times |\mathrm{Sub}\,(\ulcorner \mathcal{S} \urcorner)|$. □

References

Roberto M. Amadio and Denis Lugiez. 2000. On the Reachability Problem in Cryptographic Protocols. In *Concurrency Theory, 11th International Conference, University Park, PA, USA, August 22-25, 2000 (Lecture Notes in Computer Science)*, Vol. 1877. Springer, 380–394.

Myrto Arapinis and Marie Duflot. 2007. Bounding Messages for Free in Security Protocols. In *Foundations of Software Technology and Theoretical Computer Science, 27th International Conference, New Delhi, India (Lecture Notes in Computer Science)*, Vol. 4855. Springer, 376–387.

Alessandro Armando, Roberto Carbone, and Luca Compagna. 2014. SATMC: A SAT-Based Model Checker for Security-Critical Systems. In *Tools and Algorithms for the Construction and Analysis of Systems- 20th International Conference, TACAS (Lecture Notes in Computer Science)*, Vol. 8413. Springer, 31–45.

Tigran Avanesov. 2011. *Resolution of constraint systems for automatic composition of security-aware Web Services.* Thesis. Université Henri Poincaré - Nancy I. http://tel.archives-ouvertes.fr/tel-00641237 http://tel.archives-ouvertes.fr/tel-00641237.

Tigran Avanesov, Yannick Chevalier, Mohammed Anis Mekki, Michaël Rusinowitch, and Mathieu Turuani. 2012. Distributed Orchestration of Web Services under Security Constraints. In *Data Privacy Management and Autonomous Spontaneus Security - 6th International Workshop, DPM 2011, and 4th International Workshop, SETOP 2011, Leuven, Belgium, September 15-16, 2011, Revised Selected Papers*, Vol. 7122. Springer, 235–252.

Tigran Avanesov, Yannick Chevalier, Michael Rusinowitch, and Mathieu Turuani. 2010. Satisfiability of general intruder constraints with a set constructor. In *Risks and Security of Internet and Systems (CRiSIS), 2010 Fifth International Conference*. 1 –8. DOI: http://dx.doi.org/10.1109/CRISIS.2010.5764919

Franz Baader and Tobias Nipkow. 1998. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA.

David Basin, Sebastian Mödersheim, and Luca Viganò. 2005. Algebraic Intruder Deductions. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), 12th International Conference. (Lecture Notes in Computer Science)*, Vol. 3835. Springer, 549–564.

Anguraj Baskar, Ramaswamy Ramanujam, and S. P. Suresh. 2010. A Dexptime-Complete Dolev-Yao Theory with Distributive Encryption. In *Mathematical Foundations of Computer Science 2010, 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23-27 (Lecture Notes in Computer Science)*, Vol. 6281. Springer, 102–113.

Mathieu Baudet. 2005. Deciding security of protocols against off-line guessing attacks. In *ACM Conference on Computer and Communications Security*. ACM, 16–25.

Bruno Blanchet. 2009. Automatic verification of correspondences for security protocols. *Journal of Computer Security* 17, 4 (2009), 363–434.

Sergiu Bursuc, Hubert Comon-Lundh, and Stéphanie Delaune. 2007. Associative-commutative deducibility constraints. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS'07)*, *(Lecture Notes in Computer Science)*, Vol. 4393. Springer, 634–645.

Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. 2005. An NP decision procedure for protocol insecurity with XOR. *Theor. Comput. Sci.* 338, 1-3 (2005), 247–274.

Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. 2008. Complexity results for security protocols with Diffie-Hellman exponentiation and commuting public key encryption. *ACM Trans. Comput. Log.* 9, 4 (2008).

Yannick Chevalier, Denis Lugiez, and Michaël Rusinowitch. 2007. Towards an Automatic Analysis of Web Service Security. In *FroCoS 2007, Liverpool, UK, September 10-12 (Lecture Notes in Computer Science)*, Vol. 4720. Springer, 133–147.

Yannick Chevalier and Michaël Rusinowitch. 2005. Combining Intruder Theories. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings (Lecture Notes in Computer Science)*, Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung (Eds.), Vol. 3580. Springer, 639–651. DOI:http://dx.doi.org/10.1007/11523468_52

Yannick Chevalier and Michael Rusinowitch. 2010. Symbolic protocol analysis in the union of disjoint intruder theories: Combining decision procedures. *Theoretical Computer Science* 411, 10 (2010), 1261 – 1282. DOI:http://dx.doi.org/DOI:10.1016/j.tcs.2009.10.022 ICALP 2005 - Track C: Security and Cryptography Foundations.

Hubert Comon-Lundh. 2004. Intruder Theories. In *FoSSaCS (Lecture Notes in Computer Science)*, Vol. 2987. Springer, 1–4.

Hubert Comon-Lundh and Vitaly Shmatikov. 2003. Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive or. *Logic in Computer Science, Symposium on* 0 (2003), 271. DOI:http://dx.doi.org/10.1109/LICS.2003.1210067

Véronique Cortier and Stéphanie Delaune. 2012. Decidability and Combination Results for Two Notions of Knowledge in Security Protocols. *J. Autom. Reasoning* 48, 4 (2012), 441–487. DOI:http://dx.doi.org/10.1007/s10817-010-9208-8

Stéphanie Delaune. 2006. *Vérification des protocoles cryptographiques et propriétés algébriques*. Thèse de doctorat. Laboratoire Spécification et Vérification, ENS Cachan, France. http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/these-delaune.pdf

Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. 2008. Symbolic protocol analysis for monoidal equational theories. *Information and Computation* 206, 2-4 (Feb.-April 2008), 312–351. DOI:http://dx.doi.org/10.1016/j.ic.2007.07.005

Daniel J. Dougherty and Joshua D. Guttman. 2013. An Algebra for Symbolic Diffie-Hellman Protocol Analysis. In *Trustworthy Global Computing - 7th International Symposium, TGC 2012 (Lecture Notes in Computer Science)*, Vol. 8191. Springer, 164–181.

Serdar Erbatur, Andrew M. Marshall, Deepak Kapur, and Paliath Narendran. 2011. Unification over Distributive Exponentiation (Sub)Theories. *Journal of Automata, Languages and Combinatorics* 16, 2-4 (2011), 109–140.

Santiago Escobar, Deepak Kapur, Christopher Lynch, Catherine Meadows, José Meseguer, Paliath Narendran, and Ralf Sasse. 2011. Protocol analysis in Maude-NPA using unification modulo homomorphic encryption. In *PPDP*, Peter Schneider-Kamp and Michael Hanus (Eds.). ACM, 65–76.

Maria-Camilla Fiazza, Michele Peroli, and Luca Viganò. 2012. An environmental paradigm for defending security protocols. In *2012 International Conference on Collaboration Technologies and Systems, CTS 2012, Denver, CO, USA, May 21-25, 2012*. IEEE, 427–438.

Bogdan Groza and Marius Minea. 2013. Bridging Dolev-Yao Adversaries and Control Systems with Time-Sensitive Channels. In *Critical Information Infrastructures Security - 8th International Workshop, CRITIS 2013, Amsterdam, The Netherlands, September 16-18 (Lecture Notes in Computer Science)*, Vol. 8328. Springer, 167–178.

Joshua D. Guttman. 2007. How to do Things with Cryptographic Protocols. In *Advances in Computer Science - ASIAN 2007. Computer and Network Security, 12th Asian Computing Science Conference, Doha, Qatar, December 9-11, 2007, Proceedings (Lecture Notes in Computer Science)*, Vol. 4846. Springer.

Ali Kassem, Pascal Lafourcade, Yassine Lakhnech, and Sebastian Mödersheim. 2013. Verifiability in e-Auction Protocols & Brandt's Protocol Revisited. In *1st Workshop on Hot Issues in Security Principles and Trust (HotSpot'13)*.

Ralf Küsters and Tomasz Truderung. 2008. Reducing protocol analysis with XOR to the XOR-free case in the horn theory based approach. In *ACM Conference on Computer and Communications Security*, Peng Ning, Paul F. Syverson, and Somesh Jha (Eds.). ACM, 129–138.

Zhiqiang Liu and Christopher Lynch. 2011. Efficient General Unification for XOR with Homomorphism. In *Automated Deduction - CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw, Poland, July 31 - August 5. Proceedings (Lecture Notes in Computer Science)*, Vol. 6803. Springer, 407–421.

Christopher Lynch and Catherine Meadows. 2004. Sound Approximations to Diffie-Hellman Using Rewrite Rules. In *ICICS (Lecture Notes in Computer Science)*, Vol. 3269. Springer, 262–277.

Sreekanth Malladi. 2012. Soundness of Removing Cancellation Identities in Protocol Analysis under Exclusive-OR *(Lecture Notes in Computer Science)*, Vol. 6993. Springer, 205–224.

Laurent Mazaré. 2005. Satisfiability of Dolev-Yao Constraints. *Electronic Notes in Theoretical Computer Science* 125, 1 (2005), 109–124.

Laurent Mazaré. 2006. *Computational Soundness of Symbolic Models for Cryptographic Protocols*. Ph.D. Dissertation. Institut National Polytechnique de Grenoble. http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/these-mazare.pdf

Catherine Meadows. 2011. Theorem Proving and Security. In *Encyclopedia of Cryptography and Security (2nd Ed.)*, Henk C. A. van Tilborg and Sushil Jajodia (Eds.). Springer, 1285–1287.

Sebastian Mödersheim, Flemming Nielson, and Hanne Riis Nielson. 2013. Lazy Mobile Intruders. In *Principles of Security and Trust - Second International Conference, POST 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings (Lecture Notes in Computer Science)*, Vol. 7796. Springer, 147–166. DOI:http://dx.doi.org/10.1007/978-3-642-36830-1

OWASP Foundation. 2008. OWASP-DV-008, OWASP Testing Guide, v3.0. http://www.owasp.org/index.php/Testing_for_XML_Injection_(OWASP-DV-008). (2008). http://www.owasp.org/index.php/Testing_for_XML_Injection_(OWASP-DV-008)

Michaël Rusinowitch and Mathieu Turuani. 2003. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theor. Comput. Sci.* 1-3, 299 (2003), 451–475.

Paul Syverson, Catherine Meadows, and Iliano Cervesato. 2000. Dolev-Yao is no better than Machiavelli. In *First Workshop on Issues in the Theory of Security — WITS'00*. 87–92.

Mathieu Turuani. 2006. The CL-Atse Protocol Analyser. In *Term Rewriting and Applications (RTA). (Lecture Notes in Computer Science)*, Vol. 4098. Springer, 277–286.

Luca Viganò. 2012. Automated validation of trust and security of service-oriented architectures with the AVANTSSAR platform. In *International Conference on High Performance Computing & Simulation, HPCS 2012, Madrid*. IEEE, 444–447.