Satisfiability of General Intruder Constraints with and without a Set Constructor

Tigran Avanesov, Yannick Chevalier, Michael Rusinowitch, Mathieu Turuani

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Satisfiability of General Intruder Constraints with and without a Set Constructor

Tigran Avanesov — Yannick Chevalier — Michaël Rusinowitch — Mathieu Turuani

N° 7276

May 21, 2010

Thèmes COM et NUM et SYM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7276--FR+ENG

Satisfiability of General Intruder Constraints with and without a Set Constructor

Tigran Avanesov* , Yannick Chevalier† , Michaël Rusinowitch ‡ , Mathieu Turuani§

Thèmes COM et NUM et SYM — Systèmes communicants et Systèmes numériques et Systèmes symboliques
Projets CASSIS

Rapport de recherche n° 7276 — May 21, 2010 — 50 pages

Abstract: Many decision problems on security protocols can be reduced to solving so-called intruder constraints in the Dolev Yao model. Most constraint solving procedures for cryptographic protocol security rely on two properties of constraint systems called *knowledge monotonicity* and *variable-origination*. In this work we relax these restrictions by giving an NP decision procedure for solving general intruder constraints (that do not have these properties). Our result extends a first work by L. Mazaré in several directions: we allow non-atomic keys, and an associative, commutative and idempotent symbol (for modeling sets). We also discuss several new applications of our result.

Key-words: Security, Constraint solving, Dolev-Yao intruder, equational theory, ACI

* CASSIS, Loria-INRIA Grand Est, Tigran.Avanesov@loria.fr
† IRIT-Université Paul Sabatier, ychevali@irit.fr
‡ CASSIS, Loria-INRIA Grand Est, Michael.Rusinowitch@loria.fr
§ CASSIS, Loria-INRIA Grand Est, Mathieu.Turuani@loria.fr

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Satisfiabilité des systèmes de contraintes généraux avec et sans constructeur d'ensemble

Résumé : De nombreux problèmes de décision relatifs à la sécurité des protocoles cryptographiques peuvent être réduits à la résolution de ce que l'on appelle contraintes d'intrus dans le modèle de Dolev Yao . La plupart des procédures de résolution de contraintes pour la sécurité des protocoles se basent sur deux propriétés de ces systèmes appelées *monotonie des connaissances* et *ordonnancement des variables*. Dans ce travail nous relâchons ces restrictions en présentant une procédure de décision dans NP pour la résolution de contraintes d'intrus générales, c'est à dire non soumises aux deux propriétés précédentes. Notre résultat prolonge un premier travail de L. Mazaré dans plusieurs directions: nous autorisons les clés non-atomique, ainsi qu'un symbole associatif, commutatif et idempotent (pour modéliser les ensembles). Nous considérons également de nouvelles applications de ce résultat.

Mots-clés : Sécurité, résolution de contraintes, intrus de Dolev-Yao, théorie équationelle, ACI

Contents

1 Introduction

Detecting flaws in security protocol specifications under the perfect cryptography assumption in Dolev-Yao intruder model is an approach that has been extensively investigated in recent years [18, 2, 22, 12]. In particular symbolic constraint solving has proved to be a very successful approach in the area. It amounts to express the possibility of mounting an attack, e.g. the derivation of a secret, as a list of steps where for each step some message 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 standard Dolev-Yao intruder model with different equational theories [1, 9] like exclusive OR, modular exponentiation, Abelian groups, etc. [11, 7, 14] 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 an associative commutative and idempotent operator) since it allows one to express sets in cryptographic protocols. To our knowledge this intruder theory has not been considered before.

Up to one exception [16, 17], all proposed algorithms rely on two strong assumptions about the constraints to be processed: knowledge monotonicity and variable origination. Constraints satisfying this hypothesis are called *well-formed constraints* in the literature and they are not restrictive as these conditions hold when handling standard security problems with a single Dolev-Yao intruder. However, we will see that in some situations it can be quite useful to relax these hypothesis 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). Remark, that if intruders can communicate during protocol execution, the model becomes attack-equivalent to one with a unique Dolev-Yao intruder [21]. General constraints are also interesting for modelling cryptographic protocols where some data (e.g. initial agent knowledge) are parameterized (see § 2.2).

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 the usage of an associative commutative idempotent operator (Sections 3, 4) that can be used for instance to model sets of nodes in XML document (see § 2.3). Third, we will remark that the satisfiability procedure we obtain for *general constraints* is a non trivial extension of the one for *well-formed constraints* by showing that this procedure cannot be extended to handle operators with subterm convergent theories since satisfiability gets undecidable in this case (Appendix A). On the other hand it is known that satisfiability remains decidable for the standard case of *well-formed constraints* with the same operator properties [3]. Finally we will sketch the potential applications of our results (Section 2).

1.2 Related works

The decision procedure for satisfiability of constraint systems can be used to decide the insecurity of cryptographic protocols with a bounded number of sessions [19]. In this domain, several works deviated from perfect cryptography assumption and started to consider algebraic properties of functional symbols. For example properties of XOR operator and exponentiation were considered in [6, 7, 10, 20] and together with homomorphic symbol in [13]. Some algebraic properties (like associative and commutative symbol) make the insecurity problem undecidable [5]. Due to decidability of constraint system with ACI symbol, the problem of insecurity for protocols with ACI constructor is also decidable (under condition of bounded number of sessions). Works [13, 14, 16, 17, 11, 4] we aware of do not cover the case of general DY+ACI constraints

All the works listed above considered systems of constraints with two restrictions namely knowledge monotonicity (left-hand side of a constraint is included into left-hand side of the next one) and variable origination (variable appears first in 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 so-called "well-formed" constraints was made by Mazaré [16]. He considered "quasi well-formed" constraint systems by relaxing the knowledge monotonicity. Later, in his thesis [17], he raised a similar decidability problem, but now for general constraint systems. He succeeded to find a decision procedure for satisfiability of general constraint systems with the restriction that keys used for encryption are atomic. On the other hand, no extension of the classical Dolev-Yao deduction system was proposed for general constraint systems.

1.3 Acknowledges

The work presented in this paper was partially supported by the FP7-ICT-2007-1 Project no. 216471, "AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures" (http://www.avantssar.eu).

2 Motivating examples

2.1 Protocol analysis with several intruders

In the domain of security protocol analysis Dolev-Yao model is widely used in spite of its limitations. We propose here to consider instead of a powerful intruder that controls the whole network, non communicating several intruders with smaller controlled domains. We give below an application of this model.

Let us suppose that four agents (A, B, C and D) intend to exchange messages (see Figure 1). Some data possessed by these agents are supposed to be secret. The agents have established communication channels between each other. Due to their layout (long distance between the hosts) they have to transmit data through routers: messages sent by channel AB, i.e. from A to B, (and vice versa, from B to A) will come through Routers 1 and 2; A to C through 3, 6, 7; A to D through Router 3; B to C through 5; B to D through 4 and 7 and C to D through Router 8. Some routers are compromised (for example, the intruder managed to install a malicious pre-programmed device controlling input and output channels of every compromised router or malicious code); in our example routers under control of an intruder are 2, 5 and 7. We assume that during message exchanges compromised routers have no means to communicate: due to the network topology there is no communication link between them, neither direct, nor via other routers. After a while, an intruder can collect the knowledge of every compromised router (he can get back implanted malicious devices and read their memory or, if it was a code, copy stored data). The specification of the protocol, which agents are going to execute, is known initially by the intruder, as well as

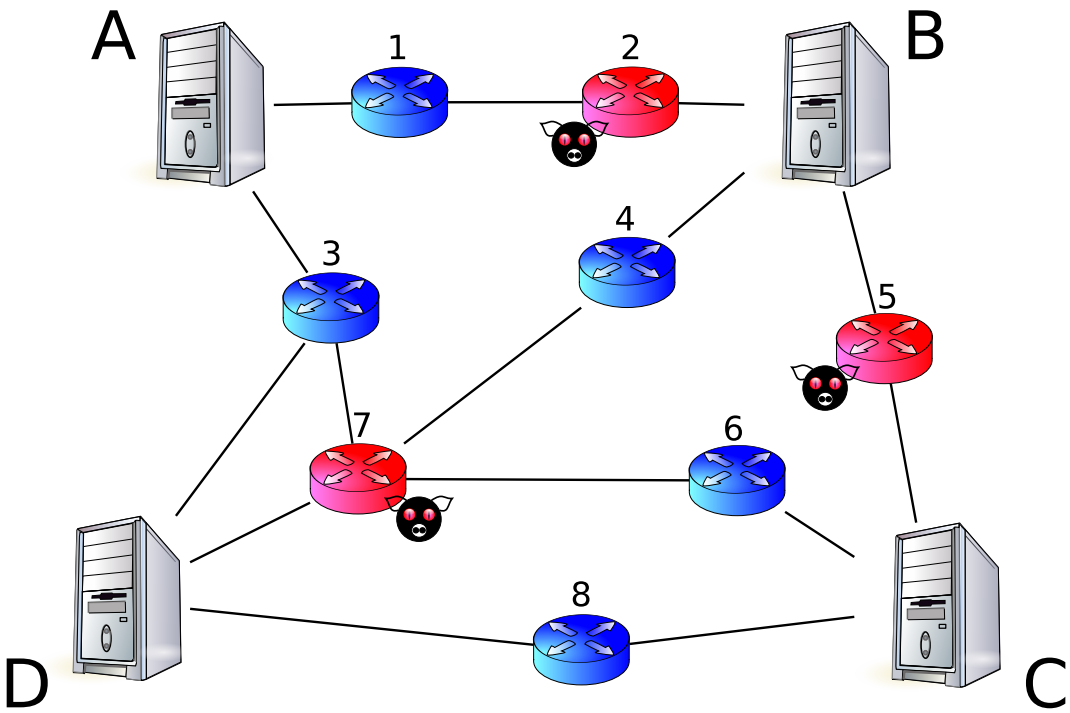

Figure 1: Untrusted routers

which agent will execute which role. So in this framework the *security problem* is to know whether it is possible for the intruder to initially give instructions to compromised routers (e.g. by reprogramming devices) to force honest agents to reveal a secret. More formally the problem is to decide whether the secret data will be deducible from the gathered knowledge of all compromised routers (devices memory) after some preprogrammed interactions between compromised routers and honest agents.

More generally the problem input is a set of agents each given with a list of actions it is supposed to execute. Every agent has communication channels with other ones. We bind to every communication channel at most one active "local" intruder (several channels can be controlled by the same intruder, some can be free from intruders). This binding is also given. Every local intruder has initial knowledge. Intruder controlling a channel can intercept a message passed on it thereby increasing his knowledge, and forge a message (anything he can build from his current knowledge) to send to the endpoint of any channel controlled by him. Local intruders cannot exchange messages during protocol execution but only afterwards. The question posed is, given a set of secret data, whether there exists a sequence of message exchanges between honest agents (which strictly follow the list of actions) and local intruders such that at some point, from common knowledge of all local intruders it is possible to deduce a secret data.

2.1.1 Formalizing the coordinated attack problem

To formalize the problem we introduce some notations and definitions. Let $\mathcal{A}$ be the set of atoms representing elementary pieces of data: texts, public keys, names of agents, etc. Let $\mathcal{X}$ be the set of variables, representing data holders, values (possibly composed) of which are asked to be found. We define a term as in Def. 3.1: classically for protocol security domain, we allow encryption, pairing, etc. Let $\mathcal{T}$ be the set of all possible terms. Let t be a term. We define $\mathrm{Vars}\,(t)$ the set of all variables in t (see Def. 3.9). We call term t a *ground term*, if $\mathrm{Vars}\,(t) = \emptyset$. The set of ground terms is denoted by $\mathcal{T}_g$. An equational theory may be considered (in our case is commutativity, associativity and idempotence of $\cdot$). Every term t is supposed to have a unique normal form denoted by $\ulcorner t \urcorner$. A term t is *normalized* if $t = \ulcorner t \urcorner$. Two terms p and q are *equivalent*, if $\ulcorner p \urcorner = \ulcorner q \urcorner$. Given a set of terms T we define $\ulcorner T \urcorner = \{\ulcorner t \urcorner : t \in T\}$. More details are given in Section 3.

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 variable x_i by the corresponding term t_i. The set of variables $\{x_1, \ldots, x_k\}$ is called the *domain* of σ and denoted by $\mathrm{dom}\,(\sigma)$. 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. We will say that the substitution σ is normalized, if $\forall x \in \mathrm{dom}\,(\sigma)$, $x\sigma$ is normalized.

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

Channels Between two any agents a and b there exists a communication channel which we denote as $a \rightharpoonup b$. We will suppose, that channels are directed. Set of all channels denoted as $\mathbb{C}$.

A channel supports a queue of messages: for example, if a sends sequentially two messages to b (via channel $a \rightharpoonup b$), then b cannot process the second message before the first one; messages are stored in queue to be processed in order of arrival.

Agents behavior We define a protocol session $PS = \{\langle a_i, l_i \rangle\}_{i=1,\ldots,k}$ as a set of pairs of an agent name and a finite list of actions to be executed by this agent[1]. We also suppose that $\mathrm{Vars}\,(l_i) \cap \mathrm{Vars}\,(l_j) = \emptyset$, for all $i \neq j$ (where $\mathrm{Vars}\,(\cdot)$ is naturally extended on lists of actions).

Every action is of type $?_f r$ or $!_t s$, where for the first form

- f is an agent name, whom a message is to be received from;
- r is a term (a template for the message) expected to be received from f;

and for the second

- t is an agent name, whom the message is expected to be sent to;
- s is a term (a template for the message) to be sent to t.

For any agent $a \in A$ participating in the protocol session PS, let $\left\langle a, \{\rho_i\}_{i=1,\ldots,k} \right\rangle \in PS$.

Case 1. If $\rho_1 = ?_{f_1} r_1$ then the first action agent a can do, is to accept a message m, admittedly from agent f_1 on channel $f_1 \rightharpoonup a$, matching the pattern r_1, i.e. such that $\ulcorner r_1 \sigma \urcorner = \ulcorner m \urcorner$ for some substitution σ. Agent is blocked (does not execute any other actions) by awaiting a message. If a receives a message that does not match the expected pattern, then a terminates his participation in PS. Note, that no notification is sent to the sender, thus a sender continues his execution[2]. Once a has received message m matching the pattern r_1 with substitution σ, a moves to a state where the list of actions to be executed is $\{\rho_i \sigma\}_{i=2,\ldots,k}$, with

$$\rho\sigma = \begin{cases} ?_f(r\sigma), & \text{if } \rho = ?_f r; \\ !_t(s\sigma), & \text{if } \rho = !_t s. \end{cases}$$

We will say that an action ρ is ground, if in case of $\rho = ?_f r$, r is a ground term; or in case of $\rho = !_t s$, s is ground.

Case 2. If ρ_1 is $!_{t_1} s_1$ then the first action of agent a is sending message s_1 to agent t_1 (i.e. putting it to channel $a \rightharpoonup t_1$) and then, moving to a state where $\{\rho_i\}_{i=2,\ldots,k}$ has to be executed.

[1] For simplicity, we suppose that for a protocol session, one agent can not have more than one list of actions to execute, but this restriction can be relaxed.

[2] One of the way to model another behavior, is to explicitly provide for every sending a succedent receive of an acknowledge message and for every receive a succedent send of an acknowledge message.

Intruder model We assume that some communication channels are controlled by *local* intruders and there is no channel controlled by two or more intruders.

Suppose, there are N local intruders. We identify intruders by their names: $\{I_i\}_{i=1,\ldots,N}$. We introduce an *intruders layout* represented by a function $\iota : \mathbb{C} \mapsto \mathbb{I} \cup \{\varnothing\}$, which gives an information on what intruder controls given channel (channel $a \rightharpoonup b$ is free of intruders iff $\iota(a \rightharpoonup b) = \varnothing$).

Every intruder I has some initial knowledge K_I^0 that is a set of ground terms.

Once an agent sends a message via a channel controlled by 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 his deduction power and send it to its endpoint.

We introduce some formalism needed to 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.*

From now to the end of this section rules are assumed to belong to a fixed deduction system $\mathcal{D}$.

Definition 2.2. *A* ground instance *of a rule $d = s_1, \ldots, s_k \to s$ is a rule $l = l_1, \ldots, l_k \to r$ where $l_1, \ldots, l_k, r$ are ground terms and there exists σ — ground substitution, such that $l_i = s_i\sigma, \forall i = 1, \ldots, k$ and $r = s\sigma$. We will also call a ground instance of a rule a* ground rule *when there is no ambiguity.*

Given two sets of ground terms E, F and a rule $l \to r$, we write $E \to_{l\to r} F$ iff $F = E \cup \{r\}$ 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 2.3. *A* derivation *D of length $n \geq 0$ is a 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 $E_i = E_{i-1} \cup \{t_i\}, \forall i = \{1, \ldots, n\}$. A term t is* derivable *from a set of terms E iff there exists a derivation $D = E_0, \ldots, E_n$ such that $E_0 = E$ and $t \in E_n$. A set of terms T is derivable from E, iff every $t \in T$ is derivable from E. We denote* $\mathrm{Der}\,(E)$ *set of terms derivable from E.*

Local intruder I can send a message m, if $m \in \mathrm{Der}\,(K_I)$, where K_I is a current knowledge of intruder I.

Protocol session execution Now, having a protocol session, intruders layout and theirs initial knowledges, we can present a course of a protocol execution. First, we will introduce an execution under some constraints. This kind of execution we will call *symbolic*. And then, once the constraints are satisfied, we deal with an effective execution.

Definition 2.4. *Let 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. *A* constraint system *is a set*

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

where n is an integer and $E_i \rhd t_i$ is a constraint for all $i \in \{1, \ldots, n\}$.

We extend the definition of $\mathrm{Vars}(\cdot)$ to constraint system $\mathcal{S}$ in a natural way. We will say that $\mathcal{S}$ is normalized, if every term occurring in $\mathcal{S}$ is normalized. As $\ulcorner\mathcal{S}\urcorner$ we will denote a constraint system $\{\ulcorner E_i \urcorner \rhd \ulcorner t_i \urcorner\}_{i=1,\ldots,n}$.

Definition 2.5. *A ground substitution σ is a* solution *of constraint $E \rhd t$ (or σ satisfies this constraint), if $\ulcorner t\sigma \urcorner \in \mathrm{Der}(\ulcorner E\sigma \urcorner)$. A ground substitution σ is a* solution *of a constraint system $\mathcal{S}$, if it satisfies all the constraints of $\mathcal{S}$ and $\mathrm{dom}(\sigma) = \mathrm{Vars}(\mathcal{S})$.*

Definition 2.6. *A configuration Π of a protocol session is a tuple $\langle PS, \mathcal{K}, \mathcal{Q}, \mathcal{S}\rangle$, where $\mathcal{K} = \{\langle I_i, K_i\rangle\}_{i=1,\ldots,N}$ represents current knowledges of intruders, and $\mathcal{Q} = \{\langle c, m_c\rangle\}_{c\in\mathbb{C}}$ is a configuration of channels: for every channel c queue of messages m_c is given.*

For any substitution σ we define

$$\left\langle \{\langle a_i, l_i\rangle\}_{i=1,\ldots,k}, \{\langle I_i, K_i\rangle\}_{i=1,\ldots,N}, \{\langle c, m_c\rangle\}_{c\in\mathbb{C}}, \{E_i \rhd t_i\}_{i=1,\ldots,n} \right\rangle \sigma$$

as

$$\left\langle \{\langle a_i, l_i\sigma\rangle\}_{i=1,\ldots,k}, \{\langle I_i, K_i\sigma\rangle\}_{i=1,\ldots,N}, \{\langle c, m_c\sigma\rangle\}_{c\in\mathbb{C}}, \{E_i\sigma \rhd t_i\sigma\}_{i=1,\ldots,n} \right\rangle$$

where application of a substitution on a list is defined in a natural way.

Then we define a set of configuration transitions in Table 1:

1.	$\langle\{\langle a, (?_f r).l_a\rangle\} \cup PS, \{\langle I, K\rangle\} \cup \mathcal{K}, \mathcal{Q}, \mathcal{S}\rangle \xrightarrow{\iota(f \rightharpoonup a)=I}$ $\langle\{\langle a, l_a\rangle\} \cup PS, \{\langle I, K\rangle\} \cup \mathcal{K}, \mathcal{Q}, \mathcal{S} \cup \{K \rhd r\}\rangle$
2.	$\langle\{\langle a, (!_t s).l_a\rangle\} \cup PS, \{\langle I, K\rangle\} \cup \mathcal{K}, \mathcal{Q}, \mathcal{S}\rangle \xrightarrow{\iota(a \rightharpoonup t)=I}$ $\langle\{\langle a, l_a\rangle\} \cup PS, \{\langle I, K \cup s\rangle\} \cup \mathcal{K}, \mathcal{Q}, \mathcal{S}\rangle$
3.	$\langle\{\langle a, (!_t s).l_a\rangle\} \cup PS, \mathcal{K}, \{\langle a \rightharpoonup t, m_{a\rightharpoonup t}\rangle\} \cup \mathcal{Q}, \mathcal{S}\rangle \xrightarrow{\iota(a \rightharpoonup t)=\varnothing}$ $\langle\{\langle a, l_a\rangle\} \cup PS, \mathcal{K}, \{\langle a \rightharpoonup t, m_{a\rightharpoonup t}.s\rangle\} \cup \mathcal{Q}, \mathcal{S}\rangle$
4.	$\langle\{\langle a, (?_f r).l_a\rangle\} \cup PS, \mathcal{K}, \{\langle f \rightharpoonup a, s.m_{f\rightharpoonup a}\rangle\} \cup \mathcal{Q}, \mathcal{S}\rangle \xrightarrow[\ulcorner s\sigma\urcorner=\ulcorner r\sigma\urcorner]{\iota(f \rightharpoonup a)=\varnothing}$ $\langle\{\langle a, l_a\rangle\} \cup PS, \mathcal{K}, \{\langle f \rightharpoonup a, m_{f\rightharpoonup a}\rangle\} \cup \mathcal{Q}, \mathcal{S}\rangle\,\sigma$

Note, that for transitions if we write $\{x\} \cup Y$ it implicitly means $x \notin Y$.

Table 1: Configuration transitions

Definition 2.7. *A sequence of configurations obtained by application of transitions to the initial configuration $\left\langle PS, \left\{\left\langle I, K_I^0\right\rangle\right\}_{I\in\mathbb{I}}, \{\langle c, \emptyset\rangle\}_{c\in\mathcal{C}}, \emptyset\right\rangle$ is a symbolic execution of protocol session PS under conditions of intruders ι.*

An *execution* $E_{PS} = \{C_i\sigma\}_{i=1,\ldots,m}$ is an instance of symbolic execution $\{C_i\}_{i=1,\ldots,m}$ (where $C_i = \langle PS_i, \mathcal{K}_i, \mathcal{Q}_i, \mathcal{S}_i\rangle$) such that all terms of $C_i\sigma$ are ground and $\mathcal{S}_m$ is satisfied by σ.

Offline communication After a protocol session is executed, the current knowledge of all local intruders will be gathered to derive a secret, which, probably, separately they cannot deduce.

Coordinated attack problem Now we can formally state the problem.
Input: Given a finite set of agents A, protocol session $PS = \{\langle a_i, l_i\rangle\}_{i=1,\dots,k}$, a set of intruders $\mathbb{I} = \{I_i\}_{i=1,\dots,N}$ with its initial knowledges $K^0_{I_i}$, intruder layout ι and sensitive data, represented by a finite set of ground terms S.
Output: Find out, whether exists $s \in S$ and an execution E_{PS} of protocol session PS with its last configuration $\langle PS, \mathcal{K}, \mathcal{Q}, \mathcal{S}\rangle$ such that $s \in \mathrm{Der}\left(\bigcup_{\langle K_I, I\rangle \in \mathcal{K}} K_I\right)$.

2.1.2 Solving the problem

We will consider another mode (let us call it *constraint mode*) of symbolic execution. Instead of Transition 4 (Table 1) we will consider the following transition:

$$\langle\{\langle a, (?_f r).l_a\rangle\} \cup PS, \mathcal{K}, \{\langle f \rightharpoonup a, s.m_{f\rightharpoonup a}\rangle\} \cup \mathcal{Q}, \mathcal{S}\rangle \xrightarrow{\iota(f\rightharpoonup a)=\varnothing}$$
$$\langle\{\langle a, l_a\rangle\} \cup PS, \mathcal{K}, \{\langle f \rightharpoonup a, m_{f\rightharpoonup a}\rangle\} \cup \mathcal{Q}, \mathcal{S} \cup \{\{\mathrm{enc}\,(s\sigma, k)\} \rhd \mathrm{enc}\,(r\sigma, k)\}\rangle$$

Given a coordinated attack problem input, a set of possible executions obtained from symbolic executions in "original mode" (symbolic execution using transitions in Table 1) coincides with a set of possible executions obtained from symbolic executions in constraint mode. This is due to the fact that $\ulcorner t_1\sigma \urcorner = \ulcorner t_2\sigma \urcorner$ iff σ is a solution of $\mathrm{enc}\,(t_1, k) \rhd \mathrm{enc}\,(t_2, k)$ (for any term k), i.e. $\ulcorner t_1\sigma \urcorner = \ulcorner t_2\sigma \urcorner \iff \ulcorner \mathrm{enc}\,(t_1, k)\,\sigma \urcorner \in \mathrm{Der}\,(\{\ulcorner \mathrm{enc}\,(t_2, k)\,\sigma \urcorner\})$. And as the question of the problem concerns to executions, a usage of two modes is equivalent, and we will consider symbolic executions of constraint mode in this paragraph.

Assume that s is the sensitive datum that one has to check whether it can be revealed by a coordinated attack.

As every execution defines a sequence of actions executed by honest agents, we can guess the length of this sequence (this number is bounded) as well as an interleaving of actions for all the agents preserving order inside every list of actions (this is also bounded): we build a list of tuples in form $\langle A_i, \rho_i\rangle$. In this way we build an execution skeleton.

Thus, the initial problem is reduced to the following: Given a sequence of pairs (agent, list of actions to be executed) $E = \{\langle A_i, \rho_i\rangle\}_{i=1,\dots,k}$, a sensitive datum s, intruders layout ι and theirs initial knowledges K^0_I. Find out, if s can be revealed using the union of the knowledge sets of all local intruders after some protocol execution.

Having this information, we build a corresponding symbolic execution in constraint mode. Let its last configuration be $\left\langle PS, \{\langle K_I, I\rangle\}_{I\in\mathbb{I}}, \mathcal{Q}, \mathcal{S}\right\rangle$. If constraint system $\mathcal{S} \cup \left\{\bigcup_{\langle K_I, I\rangle \in \mathcal{K}} K_I \rhd s\right\}$ is satisfiable, then the protocol session is unsecure. The corresponding steps are presented in Algorithm 1.

Algorithm 1: Building an execution

Input: $E = \{\langle A_i, \rho_i \rangle\}_{i=1,\ldots,k}$, ι, $\{K_I^0 \in \mathcal{T}_g\}_{I \in \mathbb{I}}$, $s \in \mathcal{T}_g$
Output: Execution of protocol session, if exists, otherwise $\perp$

Build symbolic execution $SE = \{\langle PS_i, \mathcal{K}_i, \mathcal{Q}_i, \mathcal{S}_i \rangle\}_{i=1,\ldots,k}$ (in constraint mode) corresponding to E ;
if $SE = \varnothing$ *(was not possible to build)* **then return** $\perp$;
if $\mathcal{S}_k \cup \left\{ \bigcup_{\langle K_I, I \rangle \in \mathcal{K}_k} K_I \rhd s \right\}$ *is satisfiable with some* σ **then**
    return $(SE\sigma)\delta$, where $\delta : \mathrm{Vars}\,(SE\sigma) \mapsto \{a\}$ for some $a \in \mathcal{A}$
else
    return $\perp$
end

2.2 Synthesis of cooperation protocols

Another application of our decision procedure given in Section 3 is to solve the problem of constructing agent cooperation protocols under condition of security policies.

We consider a set of agents that have to cooperate in order to achieve some given goal. We assume that the agents can exchange messages through asynchronous communications channels. Similar problems have often been addressed in previous works and solved by calling to methods from automata synthesis, AI planning, or logic programming. Our objective here is to contribute to the state of the art by solving some cases, not considered before, where the structure of messages matters and where the security policy of each agent is an additional constraint. It is a non trivial task to find a cooperation scheme, as some agents may not trust each other, they may have their own requirements to communicate, some intermediates may be required to intervene (e.g. to provide certificates).

We abstract the communicating agents by specifying them solely by their initial knowledge (what an agent knows in the beginning of the interaction) and their goals (what he wants to obtain). The agent may create a new knowledge from what he knows at some point: he may concatenate, encrypt, decrypt (if he knows the key), sign, etc. The agent ability to cooperate takes the form of sending and receiving of messages. But some restrictions are to be imposed: an agent may not accept any message, but only those with some pre-defined pattern (this expresses his policy). He can only send the messages he can create from his knowledge. And last, the agent only sends messages to entities sharing a communication channel (with appropriate direction) with him, according to the network topology.

Note, that nothing forbids us to parametrize agents initial knowledges, e.g. we can say, that agent knows something encrypted with a given key, but we will not say what exactly. And it is still up to the solution to indicate the values that instantiate the initial knowledge of an agent.

We give an instance of the problem (see Figure 2): A writer (Agent A_1) wants to publish his new book (t). There is an enterprise which besides others services, has a Publishing

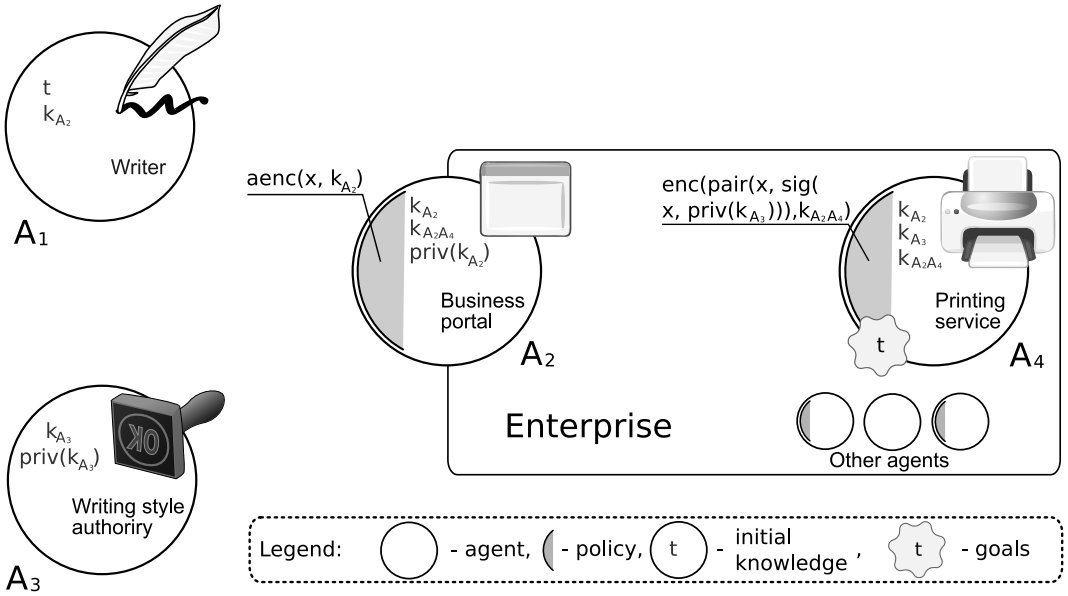

Figure 2: Illustration for agents cooperation example

Service (Agent A_4). This service accepts to print only books approved by a Writing Style Authority (Agent A_3). Anyone outside this enterprise is forbidden to access directly the Printing Service. To get access one has to contact the "Reception" (Agent A_2) of this enterprise. The Reception can communicate to the Printing Service: they share a key, and the Printing Service accepts only messages encrypted with that key.

The network topology here is as follows: A_1, A_2, A_3 are pairwise connected (as they represent public entities); A_2 and A_4 have also a communication channel (as they belong to the same enterprise).

Agent A_2 only accepts orders encrypted by his public key. Agents A_1 and A_3 can accept everything (trivial policies are omitted on Figure 2). The question is how to cooperate to print the book (A_4 should obtain t)?

This problem can also be reduced to the satisfiability of a general constraint system. For the presented example, we can obtain a non-monotonic constraint system as follows:

$$\left\{\begin{array}{l} \{t, k_{A_2}\} \rhd x_1; \{k_{A_3}, \mathrm{priv}\,(k_{A_3})\,, x_1\} \rhd x_2; \\ \{t, k_{A_2}, x_2\} \rhd \mathrm{aenc}\,(x_3, k_{A_2})\,; \\ \{k_{A_2}, k_{A_2A_4}, \mathrm{priv}\,(k_{A_4})\,, \mathrm{aenc}\,(x_3, k_{A_2})\} \rhd \mathrm{enc}\,(\mathrm{pair}\,(x_4, \mathrm{sig}\,(x_4, \mathrm{priv}\,(k_{A_3}))), k_{A_2A_4})\,; \\ \{k_{A_2}, k_{A_3}, k_{A_2A_4}, \mathrm{enc}\,(\mathrm{pair}\,(x_4, \mathrm{sig}\,(x_4, \mathrm{priv}\,(k_{A_3}))), k_{A_2A_4})\} \rhd t. \end{array}\right\}$$

and its solution is $\{x_1 \mapsto t;\ x_2 \mapsto \mathrm{sig}\,(t, \mathrm{priv}\,(k_{A_3}));\ x_3 \mapsto \mathrm{pair}\,(t, \mathrm{sig}\,(t, \mathrm{priv}\,(k_{A_3})));\ x_4 \mapsto t\}$.

2.3 Attacks based on XML-representation of messages

Here we show how to model (using our formalism) attacks based on an XML-representation of messages. A different technique to handle this kind of attacks was presented in [8].

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

It has two layers:

- the first layer exposes the list of goods for sale with their prices and process the orders by accepting payment,
- the second layer is a delivery service, that receives information from the first one about successfully paid orders, and sends the ordered goods to the buyer.

A simple scenario for ordering item is shown in Figure 3. First, a client sends an order using e-shop interface that consists of an item identifier, e-cheque, delivery address and some comments. Then, the first layer of the e-shop checks whether the price of the ordered item corresponds to the received cheque. If it does, the service uses the cheque and resends the order to the stock/delivery service (without the used e-cheque). Stock and delivery service prepare a parcel with ordered item and send 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 selected a simple pen (with ItemID simple) to buy, but she liked 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 (terms, normalization function and deduction system are defined as in § 3.1.1 except that we will write $(t_1 \cdot \ldots \cdot t_n)$ instead of $\cdot(\{t_1, \ldots, t_n\})$). Identifiers starting from 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: $\mathrm{sig}\,((ItemID \cdot Address \cdot Comments), \mathrm{priv}\,(k_s))$ — a message signed by e-shop, where k_s its public key, such that no one can produce this message except the shop. We abstract away from the procedure of checking price of the item and will suppose, that Shop Interface expects 5€ e-cheque for Item "*simple*".

We will use notation for sending and receiving as in § 2.1.

For Shop Interface we have:

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

For Shop Stock/Delivery we have:

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

Alice initially has:

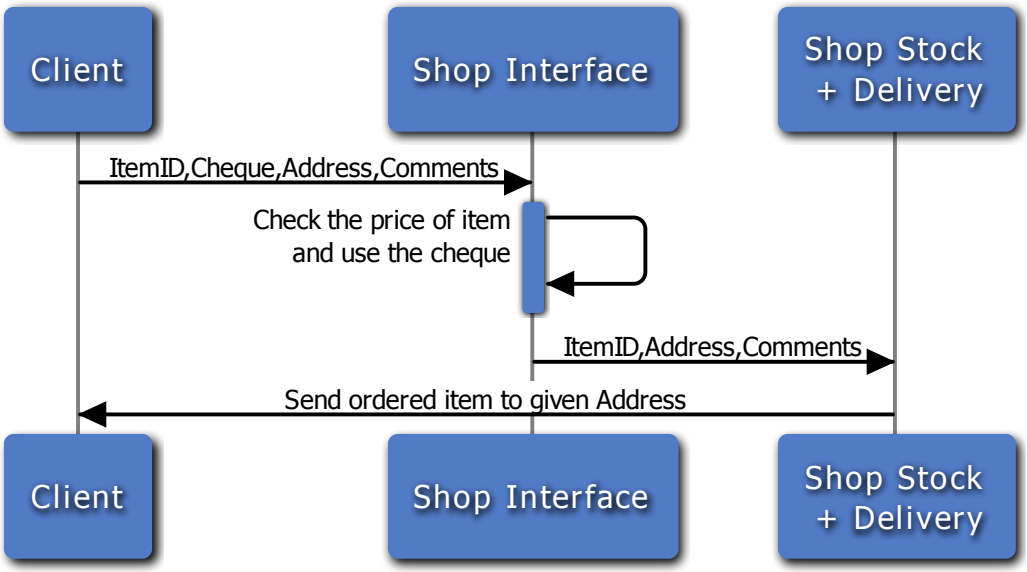

Figure 3: Ordering item scenario

$simple, gilded$**:** identifiers of items (for simplicity we suppose she knows only two of them);

$cheque5$**:** 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:

$$\left\{\begin{array}{ll} \{gilded, simple, cheque5, addr, cmnts, k_s\} \ \triangleright & \\ \qquad\qquad (simple \cdot cheque5 \cdot IAddr \cdot IComm) & (1) \\ (simple \cdot IAddr \cdot IComm) =_{ACI} (DItemID \cdot DAddr \cdot DComm) & (2) \\ \{gilded, simple, cheque5, addr, cmnts, k_s, & \\ \mathrm{sig}\,((DItemID \cdot DAddr \cdot DComm), \mathrm{priv}\,(k_s))\} \ \triangleright & \\ \qquad\qquad \mathrm{sig}\,((gilded \cdot addr \cdot DComm), \mathrm{priv}\,(k_s)) & (3) \end{array}\right\}$$

Constraint (1) shows, that Alice can construct a message expected by the shop from a client. Constraint (2) represents a request from the first to the second layer of the shop:

left-hand side is a message sent by the interface service, and right-hand side is a message expected by stock/delivery subservice. 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 most general unifier; and then of equations modulo ACI ($t_1 =_{ACI} t_2$ is equivalent to $\ulcorner t_1 \urcorner = \ulcorner t_2 \urcorner$) by encoding them into deduction rules (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 a not well-formed comments (that presents two XML-nodes), and 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 layer can return value of the first occurance of ItemID (i.e. “simple”). But the parser of the second one can return “gilded”.

This attack is possible, if Alice constructs a request “by hand”, but a similar attack is probably feasible using 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 attacks was described in [15].

3 Satisfiability of general DY+ACI constraint systems

In Section 2 we reduced the problem of protocol insecurity in presence of several intruders to solving a system of deducibility constraints. In this section we present a decision procedure

for a constraint system where Dolev-Yao deduction system is extended by an associative-commutative-idempotent symbol (DY+ACI). We consider operators for pairing, symmetric and asymmetric encryptions, decryption, signature and an ACI operator that will be used as a set constructor.

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

1. We present an algorithm for solving a ground derivability in DY+ACI model.
2. We prove, that the normalization does not change satisfiability: either we normalize a solution or a constraint system.
3. We show existence of a conservative solution of satisfiable constraint system: a substitution σ that sends a variable to an ACI-set of quasi-subterms of the constraint system instantiated with σ together with priv-ed atoms of the constraint system;
4. We give a bound on size of a conservative solution, and, as consequence, we obtain decidability.

3.1 Formal introduction to the problem

3.1.1 Terms and notions

Definition 3.1. Terms *are defined according to the following grammar:*

$$\begin{array}{ll} term & ::= variable \mid atom \mid \mathrm{pair}\,(term, term) \mid \\ & \quad \mathrm{enc}\,(term, term) \mid \cdot(tlist) \mid \mathrm{priv}\,(Keys) \mid \\ & \quad \mathrm{aenc}\,(term, Keys) \mid \mathrm{sig}\,(term, \mathrm{priv}\,(Keys)) \\ Keys & ::= variable \mid atom \\ tlist & ::= \{\ term\ [\![,\ term]\!]*\ \} \end{array}$$

where $atom \in \mathcal{A}$, $variable \in \mathcal{X}$ and $[\![x]\!]*$ stands for 0 or more repetitions of x. We denote $\mathcal{T}(\mathcal{A}, \mathcal{X})$ the set of all terms over a set of atoms $\mathcal{A}$ and a set of variables $\mathcal{X}$. For 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 $\mathrm{priv}\,(a)$ We do not assume that one can retrieve the message itself from the signature.

Note that we do allow complex keys for symmetric encryption only. As a consequence, we have to introduce a condition on substitution applications: substitution σ cannot be applied to the term t, if after replacing the resulting entity is not a term (for example, we cannot apply $\sigma = \{x \mapsto \mathrm{pair}\,(a, b)\}$ to the term $\mathrm{aenc}\,(a, x)$).

We denote a term on i-th position of a list L as $L[i]$. Then $t \in L$ is a shortcut for $\exists i : t = L[i]$. We also define two binary relations $\subseteq$ and $\approx$ on lists as follows: $L_1 \subseteq L_2$ if and only if $\forall t \in L_1 \implies t \in L_2$; $L_1 \approx L_2$ if and only if $L_1 \subseteq L_2$ and $L_2 \subseteq L_1$, and naturally extend them if L_1 or L_2 is a set.

Definition 3.2. *We consider symbol $\cdot$ to be associative, commutative, idempotent (shortly, ACI).*

We will use bin throughout the paper as a generalization of all binary operators: $\mathrm{bin} \in \{\mathrm{enc}, \mathrm{aenc}, \mathrm{pair}, \mathrm{sig}\}$.

Definition 3.3. *For every term $t \in \mathcal{T}$ we define its root symbol by*

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

Definition 3.4. *For any term $t \in \mathcal{T}$ we define its* set of elements *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. *Let us consider term* $t = \cdot\,(\{a, \cdot\,(\{b, a, \mathrm{pair}\,(a, b)\})\,, \mathrm{pair}\,(\cdot\,(\{b, b\})\,, a)\})$. *Set of its elements is* $\mathrm{elems}\,(t) = \{a, b, \mathrm{pair}\,(\cdot\,(\{b, b\})\,, a)\,, \mathrm{pair}\,(a, b)\}$

Definition 3.5. *Let $\prec$ be a strict total order on $\mathcal{T}$, such that comparing can be done in polynomial time.*

Definition 3.6. *The cardinality of a set P is denoted by $|P|$.*

Definition 3.7. *The* normal form *of a term t (denoted by $\ulcorner t \urcorner$) is recursively defined by:*

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

where for set of terms T, $\ulcorner T \urcorner = \{\ulcorner t \urcorner : t \in T\}$.

We can show easily that two terms are congruent modulo the ACI properties of $''.''$ iff they have the same normal form. Other properties are stated in Lemma 3.

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

Definition 3.8. *Let t be a term. We define a set of quasi-subterms* $\mathrm{QSub}\,(t)$ *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 — 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, we define* $\mathrm{QSub}\,(\mathcal{S}) = \bigcup_{t \in \bigcup_{i=1}^{n} E_i \cup \{t_i\}} \mathrm{QSub}\,(t)$.

Example 3. *Referring to Example 1, we have*

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

Definition 3.9. *Let t be a term. We define* $\mathrm{Vars}\,(t)$ *as set of all the variables in t:*

$$\mathrm{Vars}\,(t) = \mathcal{X} \cap \mathrm{Sub}\,(t)$$

We define $\mathrm{Sub}(t)$ as the set of subterms of t and the DAG-size of a term, as the number of its different subterms. The DAG-size gives the size of a natural representation of a term in the considered ACI theory.

Definition 3.10. *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*

$$\begin{aligned} \mathrm{Sub}\,(t) = \{\cdot\,(\{a, \cdot\,(\{b, a, \mathrm{pair}\,(a, b)\})\,, \mathrm{pair}\,(\cdot\,(\{b, b\})\,, a)\})\,, \\ \cdot\,(\{b, a, \mathrm{pair}\,(a, b)\})\,, \mathrm{pair}\,(\cdot\,(\{b, b\})\,, a)\,, \\ a, b, \mathrm{pair}\,(a, b)\,, \cdot\,(\{b, b\})\}. \end{aligned}$$

Definition 3.11. *We define a DAG-size* $\mathrm{size}_{\mathrm{DAG}}$ *of a term t as* $\mathrm{size}_{\mathrm{DAG}}\,(t) = |\mathrm{Sub}\,(t)|$, *for set of terms T,* $\mathrm{size}_{\mathrm{DAG}}\,(T) = |\mathrm{Sub}\,(T)|$ *and for constraint system $\mathcal{S}$ as* $\mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S}) = |\mathrm{Sub}\,(\mathcal{S})|$.

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

Table 2: DY+ACI deduction system rules

Remark, that for a constraint system such a definition does not polynomially approximate a number of bits needed to write it down(cf. Def. 4.1).

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 $t_1, t_2, \ldots, t_m \in \mathcal{T}$.

We suppose, hereinafter, that for a constraint system $\mathcal{S}$, $\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A} \neq \emptyset$. Otherwise, we can add one constraint $\{a\} \rhd a$ to $\mathcal{S}$ which will be satisfied by any substitution. We denote $\{\mathrm{priv}\,(t) : t \in T\}$ for set of terms T as $\mathrm{priv}\,(T)$. 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 $\forall t \in \mathrm{QSub}\,(\mathcal{S})$, t is normalized.

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

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

where $a, b, c \in \mathcal{A}$ and $x \in \mathcal{X}$.

One of the eventual solutions within DY+ACI is $\sigma = \{x \mapsto \mathrm{enc}\,(\mathrm{pair}\,(a, b)\,, c)\}$.

Definition 3.12. *Let $T = \{t_1, \ldots, t_k\}$ be a non-empty set of terms. Then we define $\pi(T)$ as follows:*

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

Remark: $\pi(\{t\}) = \ulcorner t \urcorner$.

Definition 3.13. *We denote $\mathrm{QSub}\,(\mathcal{S}) \backslash \mathcal{X}$ as $\mathrm{Q}\mathring{\mathrm{Sub}}\,(S, \mathcal{X})$ or, for shorter notation, $\mathrm{Q}\mathring{\mathrm{Sub}}\,(S)$.*

We introduce a transformation $\pi(H^{\mathcal{S},\sigma}\,(\cdot))$ on ground terms that replaces recursively all binary root symbols such that they are different from all the non-variable quasi-subterms of the constraint system instantiated with its solution σ, with ACI symbol $\cdot$. Later, we will show, that $\pi(\mathrm{H}\,(\sigma))$ is also a solution of $\mathcal{S}$.

Definition 3.14. *Let us have a constraint system $\mathcal{S}$ which is satisfiable with solution σ. Let us fix some $\alpha \in (\mathcal{A} \cap \mathrm{QSub}\,(\mathcal{S}))$. For given $\mathcal{S}$ and σ we define a function $H^{\mathcal{S},\sigma}\,(\cdot) : \mathcal{T}_g \to 2^{\mathcal{T}_g}$ as follows:*

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

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

Definition 3.15. *We define the superposition of $\pi(\cdot)$ and $H(\cdot)$ on a set of terms $T = \{t_1, \ldots, t_k\}$ as follows: $\pi(H(T)) = \{\pi(H(t)) \mid t \in T\}$.*

Definition 3.16. *Let $\theta = \{x_1 \mapsto t_1, \ldots, x_k \mapsto t_k\}$ be a substitution. We define $\pi(H(\theta))$ the substitution $\{x_1 \mapsto \pi(H(t_1)), \ldots, x_k \mapsto \pi(H(t_k))\}$.*

Note, that $\mathrm{dom}(\pi(H(\theta))) = \mathrm{dom}(\theta)$.

Example 6. *We refer to Example 5 and show, that $\pi(H(\sigma))$ is also a solution of $\mathcal{S}$. We have $\pi(H(\mathrm{enc}(\mathrm{pair}(a,b), c))) = \pi(H(\mathrm{pair}(a,b)) \cup \{c\}) = \pi(\{a\} \cup \{b\} \cup \{c\}) = \cdot(\{a,b,c\})$ (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 solution of $\mathcal{S}$ within DY+ACI.*

3.1.2 General properties used in proof

The two following lemmas state simple properties of derivability.

Lemma 1. *Let $A, B, C \subseteq \mathcal{T}_g$. Then if $A \subseteq \mathrm{Der}(B)$ and $B \subseteq \mathrm{Der}(C)$ then $A \subseteq \mathrm{Der}(C)$.*

Lemma 2. *Let $A, B, C, D \subseteq \mathcal{T}_g$. Then if $A \subseteq \mathrm{Der}(B)$ and $C \subseteq \mathrm{Der}(D)$ then $A \cup C \subseteq \mathrm{Der}(B \cup D)$.*

In Lemma 3 we list some auxiliary properties that will be used in main proof.

Lemma 3. *The following statements are true:*

1. $\ulcorner \cdot(t,t) \urcorner = \ulcorner t \urcorner$, $\ulcorner \cdot(t_1,t_2) \urcorner = \ulcorner \cdot(t_2,t_1) \urcorner$, $\ulcorner \cdot(\cdot(t_1,t_2), t_3) \urcorner = \ulcorner \cdot(t_1, \cdot(t_2,t_3)) \urcorner =$ $= \ulcorner \cdot(t_1,t_2,t_3) \urcorner$
2. *if t and $t\sigma$ are terms, then* $\ulcorner t\sigma \urcorner = \ulcorner\ulcorner t\sigma \urcorner\urcorner = \ulcorner\ulcorner t \urcorner \sigma \urcorner = \ulcorner t \ulcorner \sigma \urcorner\urcorner = \ulcorner\ulcorner t \urcorner \ulcorner \sigma \urcorner\urcorner$
3. $s \in \mathrm{QSub}(\ulcorner t \urcorner) \implies s = \ulcorner s \urcorner$
4. $\ulcorner \mathrm{elems}(t) \urcorner = \mathrm{elems}(\ulcorner t \urcorner)$

5. $\ulcorner \cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner) \urcorner = \ulcorner \cdot (t_1, \ldots, t_m) \urcorner$; $\pi(T) = \pi(\ulcorner T \urcorner)$
6. $\mathrm{elems}\,(\ulcorner \cdot (\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner) \urcorner) = \mathrm{elems}\,(\cdot\,(\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner)) = \bigcup_{i=1,\ldots,m} \mathrm{elems}\,(\ulcorner t_i \urcorner)$
7. $H\,(t) = \bigcup_{p \in \mathrm{elems}(t)} H\,(p)$,
8. $H\,(t) = H\,(\ulcorner t \urcorner)$
9. $\pi(H\,(t)) = \pi(H\,(\ulcorner t \urcorner)) = \ulcorner \pi(H\,(t)) \urcorner = \ulcorner \pi(H\,(\ulcorner t \urcorner)) \urcorner$
10. $\pi(T_1 \cup \cdots \cup T_m) = \pi(\{\pi(T_1), \ldots, \pi(T_m)\})$
11. $\mathrm{QSub}\,(\ulcorner t \urcorner) \subseteq \ulcorner \mathrm{QSub}\,(t) \urcorner$
12. $\mathrm{QSub}\,(t\sigma) \subseteq \mathrm{QSub}\,(t)\,\sigma \cup \mathrm{QSub}\,(\mathrm{Vars}\,(t)\,\sigma)$
13. $\mathrm{Sub}\,(t\sigma) = \mathrm{Sub}\,(t)\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(t)\,\sigma)$
14. $|\ulcorner T \urcorner| \leq |T|$, $|T\sigma| \leq |T|$
15. $\mathrm{elems}\,(t) \subseteq \mathrm{QSub}\,(t) \subseteq \mathrm{Sub}\,(t)$
16. *For term* t, $\mathrm{size}_{\mathrm{DAG}}\,(\ulcorner t \urcorner) \leq \mathrm{size}_{\mathrm{DAG}}\,(t)$; *for set of terms* T, $\mathrm{size}_{\mathrm{DAG}}\,(\ulcorner T \urcorner) \leq \mathrm{size}_{\mathrm{DAG}}\,(T)$; *for constraint system* $\mathcal{S}$, $\mathrm{size}_{\mathrm{DAG}}\,(\ulcorner \mathcal{S} \urcorner) \leq \mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S})$
17. $\mathrm{QSub}\,(\cdot\,(t_1, \ldots, t_l)) \subseteq \{\cdot\,(t_1, \ldots, t_l)\} \cup \mathrm{QSub}\,(t_1) \cdots \cup \mathrm{QSub}\,(t_l)$
18. $\forall s \in \mathrm{Sub}\,(t)\ \ \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner t\sigma \urcorner) \geq \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner s\sigma \urcorner)$.

Proof. We will give proofs of several statements.

Statement 1: Follows from the definition of the normalization function and Definition 3.4.

Statement 3: By induction on $\mathrm{size}_{\mathrm{DAG}}\,(s)$. Let us fix t.

- $\mathrm{size}_{\mathrm{DAG}}\,(s) = \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner t \urcorner)$. Then $s = \ulcorner t \urcorner$ and $\ulcorner s \urcorner = \ulcorner \ulcorner t \urcorner \urcorner$, and from Statement 2 (by taking empty σ) we have $\ulcorner t \urcorner = \ulcorner \ulcorner t \urcorner \urcorner$, and thus $\ulcorner s \urcorner = s$.
- Suppose, that for some k, for any $s \in \mathrm{QSub}\,(\ulcorner t \urcorner)$, such that $\mathrm{size}_{\mathrm{DAG}}\,(s) > k$, $s = \ulcorner s \urcorner$.
- Consider case, where $s \in \mathrm{QSub}\,(\ulcorner t \urcorner)$ and $\mathrm{size}_{\mathrm{DAG}}\,(s) = k$. Then, by definition of $\mathrm{QSub}\,(\cdot)$, s is in
  - $\mathrm{priv}\,(s) \in \mathrm{QSub}\,(\ulcorner t \urcorner)$. By induction supposition we have $\ulcorner \mathrm{priv}\,(s) \urcorner = \mathrm{priv}\,(s)$, and as $\ulcorner \mathrm{priv}\,(s) \urcorner = \mathrm{priv}\,(\ulcorner s \urcorner)$, we have $s = \ulcorner s \urcorner$.
  - $\mathrm{bin}\,(s, p) \in \mathrm{QSub}\,(\ulcorner t \urcorner)$. By induction we have $\ulcorner \mathrm{bin}\,(s, p) \urcorner = \mathrm{bin}\,(s, p)$, and as $\ulcorner \mathrm{bin}\,(s, p) \urcorner = \mathrm{bin}\,(\ulcorner s \urcorner, \ulcorner p \urcorner)$, we have $s = \ulcorner s \urcorner$.
  - $\mathrm{bin}\,(p, s) \in \mathrm{QSub}\,(\ulcorner t \urcorner)$. The similar case.

- $s \in \mathrm{elems}\,(\cdot\,(L))$,$\cdot\,(L) \in \mathrm{QSub}\,(\ulcorner t \urcorner)$. As $\mathrm{size}_{\mathrm{DAG}}\,(\cdot\,(L)) > k$, we have $\cdot\,(L) = \ulcorner \cdot\,(L) \urcorner$, that means (from Definition 3.7), that L is a list of normalized non-ACI-set terms, and as $\mathrm{elems}\,(L) \approx L$, we have that s is normalized.

Statement 4: This statement is trivial, if $t \neq \cdot\,(L)$. Otherwise, let $t = \cdot\,(t_1, \ldots, t_n)$.

- if $\ulcorner \mathrm{elems}\,(t) \urcorner = \{p\}$, where $p \neq \cdot\,(L_p)$. Then $\ulcorner t \urcorner = p$ and then $\mathrm{elems}\,(\ulcorner t \urcorner) = \mathrm{elems}\,(p) = \{p\} = \ulcorner \mathrm{elems}\,(t) \urcorner$.
- 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 $\mathrm{elems}\,(\ulcorner t \urcorner) = \bigcup_{p \in \{p_1, \ldots, p_k\}} \mathrm{elems}\,(p) = \{p_1, \ldots, p_k\}$.

Statement 5: The first part follows from the definition of normal form and Statement 4. The second one directly follows from the first.

Statement 6: We get the first part of equality if apply Statement 4:
$\mathrm{elems}\,(\ulcorner \cdot\,(\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner) \urcorner) = \ulcorner \mathrm{elems}\,(\cdot\,(\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner)) \urcorner$ and then from Definition 3.4 and Statement 4 we have that $\mathrm{elems}\,(\cdot\,(\ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner))$ is a set of normalized terms. The second part directly follows from Definition 3.4.

Statement 7: By induction on $\mathrm{size}_{\mathrm{DAG}}\,(t)$.

- $\mathrm{size}_{\mathrm{DAG}}\,(t) = 1$, implies $t = a \in \mathcal{A}$ and then $\mathrm{elems}\,(a) = \{a\}$, i.e. the equality becomes trivial.
- Suppose, that for any $t : \mathrm{size}_{\mathrm{DAG}}\,(t) < k$ $(k > 1)$, $H\,(t) = \bigcup_{p \in \mathrm{elems}(t)} H\,(p)$ holds.
- Given a term $t : \mathrm{size}_{\mathrm{DAG}}\,(t) = k$, $k > 1$. We shold 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 $\forall s \in L$, $\mathrm{size}_{\mathrm{DAG}}\,(s) < k$. Then, on one hand, $H\,(\cdot\,(L)) = \bigcup_{p \in L} H\,(p) =$ (by induction supposition) $= \bigcup_{p \in L} \bigcup_{p' \in \mathrm{elems}(p)} H\,(p')$. 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, $H\,(t) = \bigcup_{p \in \mathrm{elems}(t)} H\,(p)$.

Statement 8: By induction on $\mathrm{size}_{\mathrm{DAG}}\,(t)$:

- $\mathrm{size}_{\mathrm{DAG}}\,(t) = 1$ 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 $t : \mathrm{size}_{\mathrm{DAG}}\,(t) < k$ $(k > 1)$, $H\,(t) = H\,(\ulcorner t \urcorner)$ holds.
- Given a term $t : \mathrm{size}_{\mathrm{DAG}}\,(t) = k$, $k > 1$. 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 supposition) $= \{\mathrm{priv}\,(\pi(H\,(\ulcorner t_1 \urcorner)))\} = H\,(\mathrm{priv}\,(\ulcorner t_1 \urcorner)) = H\,(\ulcorner t \urcorner)$.

- if $t = \mathrm{bin}\,(p, q)$ and $\ulcorner t \urcorner \in \ulcorner \mathrm{Q}\mathring{\mathrm{S}}\mathrm{ub}\,(\mathcal{S})\,\sigma \urcorner$. 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 supposition)
  $= \{\mathrm{bin}\,(\pi(\ulcorner H\,(p) \urcorner), \pi(\ulcorner H\,(q) \urcorner))\} =$ (by Statement 5)
  $= \{\mathrm{bin}\,(\pi(H\,(p)), \pi(H\,(q)))\} = H\,(\mathrm{bin}\,(p, q))$.
- if $t = \mathrm{bin}\,(p, q)$ and $\ulcorner t \urcorner \notin \ulcorner \mathrm{Q}\mathring{\mathrm{S}}\mathrm{ub}\,(\mathcal{S})\,\sigma \urcorner$. Then $H\,(t) = H\,(p) \cup H\,(q) =$ (by induction supposition) $= H\,(\ulcorner p \urcorner) \cup H\,(\ulcorner q \urcorner) =$ (as $\ulcorner \mathrm{bin}\,(\ulcorner p \urcorner, \ulcorner q \urcorner) \urcorner = \ulcorner t \urcorner \notin \ulcorner \mathrm{Q}\mathring{\mathrm{S}}\mathrm{ub}\,(\mathcal{S})\,\sigma \urcorner$) $= H\,(\mathrm{bin}\,(\ulcorner p \urcorner, \ulcorner q \urcorner)) = H\,(\ulcorner t \urcorner)$
- if $t = \cdot\,(L)$, where $L = \{t_1, \ldots, t_m\}$. Note first, that as $t = \cdot\,(L)$, $\forall s \in \mathrm{elems}\,(t)$, $\mathrm{size}_{\mathrm{DAG}}\,(s) < \mathrm{size}_{\mathrm{DAG}}\,(t)$. Then, by Statement 7, $H\,(t) =$
  $= \bigcup_{p \in \mathrm{elems}(t)} H\,(p) =$ (by induction supposition) $= \bigcup_{p \in \mathrm{elems}(t)} H\,(\ulcorner p \urcorner)$. On the other part, $H\,(\ulcorner t \urcorner) = \bigcup_{p \in \mathrm{elems}(\ulcorner t \urcorner)} H\,(p) =$ (by Statement 4)
  $= \bigcup_{p \in \ulcorner \mathrm{elems}(t) \urcorner} H\,(p) = \bigcup_{p \in \mathrm{elems}(t)} H\,(\ulcorner p \urcorner) = H\,(t)$.

Statement 9: This follows from Statements 8, 5, Definition 3.12 and from equality $\ulcorner \ulcorner t \urcorner \urcorner = \ulcorner t \urcorner$ (Statement 2).

Statement 10: From definition of π and Statement 4, we have $\mathrm{elems}\,(\pi(T_i)) = \ulcorner \mathrm{elems}\,(T_i) \urcorner$. Next $\pi(\{\pi(T_1), \ldots, \pi(T_m)\}) = \ulcorner \cdot\,(L) \urcorner$ (here we use $\ulcorner \cdot\,(L) \urcorner$ to capture two cases from definition of normalization at once), where $L \approx \ulcorner \mathrm{elems}\,(\{\pi(T_1), \ldots, \pi(T_m)\}) \urcorner =$
$= \ulcorner \bigcup_{i=1,\ldots,m} \ulcorner \mathrm{elems}\,(T_i) \urcorner \urcorner = \ulcorner \bigcup_{i=1,\ldots,m} \mathrm{elems}\,(T_i) \urcorner$, while $\pi(T_1 \cup \cdots \cup T_m) = \ulcorner \cdot\,(L') \urcorner$, where $L' \approx \ulcorner \bigcup_{i=1,\ldots,m} \mathrm{elems}\,(T_i) \urcorner$.

Statement 11: By induction on $\mathrm{size}_{\mathrm{DAG}}\,(t)$.

- $\mathrm{size}_{\mathrm{DAG}}\,(t) = 1$. 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 $t : \mathrm{size}_{\mathrm{DAG}}\,(t) < k$ $(k > 1)$, the statement is true.
- Given a term $t : \mathrm{size}_{\mathrm{DAG}}\,(t) = k$, $k > 1$. Let us consider all possible cases:
  - $t = \mathrm{bin}\,(t_1, t_2)$.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 $\mathrm{QSub}\,(\ulcorner t_i \urcorner) \subseteq \ulcorner \mathrm{QSub}\,(t_i) \urcorner$, we have that $\mathrm{QSub}\,(\ulcorner t \urcorner) \subseteq \ulcorner \mathrm{QSub}\,(t) \urcorner$.
  - $t = \mathrm{priv}\,(t_1)$. Proof is similar to one for the case above.
  - $t = \cdot\,(L)$. We have $\mathrm{QSub}\,(t) = \{t\} \cup \bigcup_{p \in \mathrm{elems}(L)} \mathrm{QSub}\,(p)$. From defenition of normalization function we have $\mathrm{elems}\,(\ulcorner \cdot\,(L) \urcorner) = \ulcorner \mathrm{elems}\,(\cdot\,(L)) \urcorner$, and then, $\mathrm{QSub}\,(\ulcorner \cdot\,(L) \urcorner) = \{\ulcorner \cdot\,(L) \urcorner\} \cup \bigcup_{p \in \mathrm{elems}(\ulcorner \cdot(L) \urcorner)} \mathrm{QSub}\,(p) = \ulcorner \{\cdot\,(L)\} \urcorner \cup \bigcup_{p \in \mathrm{elems}(\cdot(L))} \mathrm{QSub}\,(\ulcorner p \urcorner) \subseteq$ (by supposition)
    $\subseteq \ulcorner \{\cdot\,(L)\} \urcorner \cup \bigcup_{p \in \mathrm{elems}(\cdot(L))} \ulcorner \mathrm{QSub}\,(p) \urcorner = \ulcorner \{\cdot\,(L)\} \cup \bigcup_{p \in \mathrm{elems}(\cdot(L))} \mathrm{QSub}\,(p) \urcorner = \ulcorner \mathrm{QSub}\,(t) \urcorner$.

Statement 13: By induction on $\mathrm{size}_{\mathrm{DAG}}\,(t)$

- $\mathrm{size}_{\mathrm{DAG}}\,(t) = 1$.

- $t \in \mathcal{A}$. 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 $t : \mathrm{size}_{\mathrm{DAG}}\,(t) < k$ $(k > 1)$, the statement is true.
- Given a term $t : \mathrm{size}_{\mathrm{DAG}}\,(t) = k$, $k > 1$. 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 $\mathrm{Vars}\,(t) = \mathrm{Vars}\,(t_1) \cup \mathrm{Vars}\,(t_2)$. $\mathrm{Sub}\,(t\sigma) = \{t\sigma\} \cup \mathrm{Sub}\,(t_1\sigma) \cup \mathrm{Sub}\,(t_2\sigma) =$ (as $\mathrm{size}_{\mathrm{DAG}}\,(t_i) < k$) $= \{t\sigma\} \cup \mathrm{Sub}\,(t_1)\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(t_1)\,\sigma) \cup \mathrm{Sub}\,(t_2)\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(t_2)\,\sigma) = \{t\sigma\} \cup \mathrm{Sub}\,(t_1)\,\sigma \cup \mathrm{Sub}\,(t_2)\,\sigma \cup \mathrm{Sub}\,((\mathrm{Vars}\,(t_1) \cup \mathrm{Vars}\,(t_2))\sigma) = \mathrm{Sub}\,(t)\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(t)\,\sigma)$.
  - $t = \mathrm{priv}\,(t_1)$. Proof is similar to one for case above.
  - $t = \cdot\,(\{t_1, \ldots, t_m\})$.
    We have $t\sigma = \cdot\,(\{t_1\sigma, \ldots, t_m\sigma\})$ and $\mathrm{Vars}\,(t) = \bigcup_{i=1,\ldots,m} \mathrm{Vars}\,(t_i)$. Then we have $\mathrm{Sub}\,(t\sigma) = \{t\sigma\} \cup \bigcup_{i=1,\ldots,m} \mathrm{Sub}\,(t_i\sigma) =$ (as $\mathrm{size}_{\mathrm{DAG}}\,(t_i) < k$) $= \{t\sigma\} \cup \bigcup_{i=1,\ldots,m} (\mathrm{Sub}\,(t_i)\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(t_i)\,\sigma)) = \{t\sigma\} \cup \bigcup_{i=1,\ldots,m} \mathrm{Sub}\,(t_i)\,\sigma \cup \mathrm{Sub}\left(\left(\bigcup_{i=1,\ldots,m} \mathrm{Vars}\,(t_i)\right)\sigma\right) = \mathrm{Sub}\,(t)\,\sigma \cup \mathrm{Sub}\,(\mathrm{Vars}\,(t)\,\sigma)$.

□

Lemma 4. *Given a constraint system $\mathcal{S}$ and its solution σ. Then substitution $\pi(\mathrm{H}\,(\sigma))$ is normalized*

Proof. For any $x \in \mathrm{dom}\,(\pi(\mathrm{H}\,(\sigma)))$, $x\,\pi(\mathrm{H}\,(\sigma)) = \pi(H\,(x\sigma)) = \ulcorner\pi(H\,(x\sigma))\urcorner$ (by Lemma 3). □

Lemma 5. *For any normalized term t,* $\mathrm{QSub}\,(t) = \mathrm{Sub}\,(t)$.

Proof. By induction on $\mathrm{size}_{\mathrm{DAG}}\,(t)$.

- $\mathrm{size}_{\mathrm{DAG}}\,(t) = 1$. Then $t \in \mathcal{X} \cup \mathcal{A}$, and thus, $\mathrm{QSub}\,(t) = \mathrm{Sub}\,(t) = \{t\}$.
- Suppose, that for any $t : \mathrm{size}_{\mathrm{DAG}}\,(t) < k$ $(k > 1)$, $\mathrm{QSub}\,(t) = \mathrm{Sub}\,(t)$.
- Given a term $t : \mathrm{size}_{\mathrm{DAG}}\,(t) = k$, $k > 1$. We need to show that $\mathrm{QSub}\,(t) = \mathrm{Sub}\,(t)$.
  - $t = \mathrm{bin}\,(t_1, t_2)$. Then $\mathrm{QSub}\,(\mathrm{bin}\,(t_1, t_2)) = \{t\} \cup \mathrm{QSub}\,(t_1) \cup \mathrm{QSub}\,(t_2) =$ (as $\mathrm{size}_{\mathrm{DAG}}\,(t_i) < k$) $= \{t\} \cup \mathrm{Sub}\,(t_1) \cup \mathrm{Sub}\,(t_2) = \mathrm{Sub}\,(t)$
  - $t = \mathrm{priv}\,(t_1)$. 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, $\forall p \in L$, $p \neq \cdot\,(L_p)$. Then $\mathrm{elems}\,(L) \approx L$. Thus, we have $\mathrm{QSub}\,(t) = \{t\} \cup \bigcup_{p \in \mathrm{elems}(L)} \mathrm{QSub}\,(p) = \{t\} \cup \bigcup_{p \in L} \mathrm{QSub}\,(p) = \{t\} \cup \bigcup_{p \in L} \mathrm{Sub}\,(p) = \mathrm{Sub}\,(t)$.

□

In Proposition 1 we remark, that ACI-set of normalized terms has the same deductive expressiveness as that set of normalized terms itself.

Proposition 1. *Let T be a set of terms $T = \{t_1, \ldots, t_k\}$. Then $\pi(T) \in \mathrm{Der}\,(\ulcorner T \urcorner)$ and $\ulcorner T \urcorner \subseteq \mathrm{Der}\,(\{\pi(T)\})$.*

In Proposition 2 we state that a constraint system and its normal form have the same solutions. In Proposition 3 we show the equivalence, for a constraint system, between the existence of a solution and the existence of a normalized solution. As a consequence we will need only to consider normalized constraints and solutions in the sequel.

Proposition 2. *The substitution σ is a solution of constraint system $\mathcal{S}$ if and only if σ is a solution of $\ulcorner \mathcal{S} \urcorner$.*

Proof. By definition, σ is a solution of $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$, iff $\forall i \in \{1, \ldots, n\}, \ulcorner t_i \sigma \urcorner \in \mathrm{Der}\,(\ulcorner E_i \sigma \urcorner)$. But by Lemma 3 we have 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 solution of $\mathcal{S}$ if and only if σ is a solution of $\ulcorner \mathcal{S} \urcorner$. □

Proposition 3. *The substitution σ is a solution of constraint system $\mathcal{S}$ if and only if $\ulcorner \sigma \urcorner$ is a solution of $\mathcal{S}$.*

Proof. Proof is similar to one of Proposition 2. □

3.2 Ground case of DY+ACI

In Algorithm 3 we need to check whether a ground substitution σ satisfies a constraint system $\mathcal{S}$. For this, we have to check the derivability of a ground term from a set of ground terms. In this subsection we present such an algorithm.

First, for the ground case we consider an equivalent to DY+ACI deduction system DY+ACI' obtained from the first by replacing a set of rules

$$\forall i \;\; \cdot(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$$

with

$$\forall s \in \mathrm{elems}\,(t) \;\; t \to \ulcorner s \urcorner, \text{ if } t = \cdot(L)\,.$$

Now, we show an equivalence of the two deduction systems.

Lemma 6. $t \in \mathrm{Der}_{DY+ACI}\,(E) \iff t \in \mathrm{Der}_{DY+ACI'}\,(E)$

Proof sketch. We show that every rule of one deduction system can be simulated by a combination of rules from the other. It is sufficient to show it for non common rules.

The DY+ACI' rules $\forall s \in \mathrm{elems}\,(t) \;\; t \to \ulcorner s \urcorner$, if $t = \cdot(L)$ are modeled by succesive application of rules $\forall i \;\; \cdot(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$. The converse simulation of $\cdot(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$ by DY+ACI' is based on getting all the normalized elements of t_i and, if $|\ulcorner \mathrm{elems}\,(t_i) \urcorner| \geq 2$ then reconstructing $\ulcorner t_i \urcorner$ by rule $p_1, \ldots, p_l \to \ulcorner \cdot(p_1, \ldots, p_l) \urcorner$, where $p_1, \ldots, p_l$ are $\ulcorner \mathrm{elems}\,(t_i) \urcorner$. □

Algorithm 2: Verifying derivability of term

Input: A normalized ground constraint $E \rhd t$
Output: $t \in \mathrm{Der}_{DY+ACI}(E)$

```
Let S := QSub(E) ∪ QSub(t) \ E;
Let D := E;
while true do
    if exists DY rule l → r, such that l ⊆ D and r ∈ S then
        S := S \ {r};
        D := D ∪ {r};
    else
        if exists s ∈ S : elems(s) ⊆ D then
            S := S \ {s};
            D := D ∪ {s};
        else
            if exists s ∈ D : elems(s) ⊈ D then
                S := S \ elems(s);
                D := D ∪ elems(s);
            else
                return t ∈ D;
            end
        end
    end
end
```

Lemma 7. *For Algorithm 2 the following statements are true:*

- *for any step*[3]*,* $D \cup S = \mathrm{QSub}(E \cup \{t\})$ *and* $D \cap S = \emptyset$*;*
- *it terminates;*
- *for any step,* $D \subseteq \mathrm{Der}_{DY+ACI}(E)$*.*

The following lemmas will be used to prove correctness of the algorithm.

Lemma 8.

- *For any decomposition rule* $l \to r$ *of DY+ACI', if* l *is normalized, then* r *is a quasi-subterm of* l*.*
- *For any composition rule* $l \to r$ *of DY+ACI' except* $\{t_1, \ldots, t_m\} \to \ulcorner \cdot (t_1, \ldots, t_m) \urcorner$*, if* l *is normalized, then* $l \subseteq \mathrm{QSub}(r)$*.*

[3]Consider two sequential assignments as one step

Lemma 9. *After the execution of Step 16 of Algorithm 2, if $l \to r$ is a DY+ACI' rule, such that $l \subseteq D$ and $r \notin D$, then $l \to r$ is a composition rule and $r \notin \mathrm{QSub}(E \cup \{t\})$.*

Proof. Suppose, $l \to r$ is a decomposition. By Lemma 8 we have that $r \in \mathrm{QSub}(l)$ and thus, $r \in \mathrm{QSub}(D) \subseteq D \cup S$. Then $r \notin D$ implies $r \in S$, and then, Step 16 must be skipped, as branch 4 or 12 should have been visited.

Thus, $l \to r$ is a composition. As algorithm reached Step 16, that means $r \notin S$ (otherwise one of three branches must be visited and this step would be skipped). As $r \notin S$ and $r \notin D$, we have $r \notin S \cup D = \mathrm{QSub}(E \cup \{t\})$. □

Lemma 10. *Given a set of normalized terms S such that for any $s \in S$, $\mathrm{elems}(s) \subseteq S$. Then for any DY+ACI' composition rule $l \to r$ such that $l \subseteq S$ we have $\mathrm{elems}(r) \subseteq S \cup \{r\}$.*

Proof. All cases of composition rules except $t_1, \ldots, t_m \to \ulcorner \cdot (t_1, \ldots, t_m) \urcorner$ are trivial, as for them $\mathrm{elems}(r) = \{r\}$. For this case, as $\mathrm{elems}(t_i) \subseteq S$ for all i, then (by Lemma 3, statement 6) $\mathrm{elems}(\ulcorner \cdot (t_1, \ldots, t_m) \urcorner) = \mathrm{elems}(\cdot (t_1, \ldots, t_m)) = \bigcup_{i=1}^{m} \mathrm{elems}(t_i) \subseteq S$. □

Proposition 4. *Algorithm 2 is correct.*

Proof. If algorithm returns true, then by Lemma 7 we have $t \in \mathrm{Der}_{DY+ACI'}(E)$.

Show, that output is correct, if algorithm returns false. Note, that we consider values of D and S that they have after finishing the algorithm. Suppose that output is false ($t \notin D$), but $t \in \mathrm{Der}_{DY+ACI'}(E)$. Then there exists minimal by length derivation $\{E_i\}_{i=0,\ldots,n}$ where $n \geq 1$, $D = E_0$ (as $D \subseteq \mathrm{Der}_{DY+ACI'}(E)$ and $t \notin D$) and $t \in E_n$ and $E_{i+1} \setminus E_i \neq \emptyset$ and $E_i \to_{l_i \to r_i} E_{i+1}$ for all $i = 0, \ldots, n-1$. Then, applying Lemma 9 we have $l_0 \to r_0$ is a composition, and $r_0 \notin \mathrm{QSub}(E \cup \{t\})$.

Let m be the smallest index such that there exists $s \in S = \mathrm{QSub}(E \cup \{t\}) \setminus D$ and $s \in E_m$.

Let k be the minimal integer, such that $l_k \to r_k$ is a decomposition.

Show, $k \leq m$. Suppose the opposite, then s is built by a chain of composition rules from D. If $l_{m-1} \to r_{m-1}$ (where $r_{m-1} = s$) is

- a rule in form of $\{t_1, \ldots, t_c\} \to \ulcorner \cdot (t_1, \ldots, t_c) \urcorner$, then $\mathrm{elems}(s) \neq \{s\}$ (otherwise it contradicts to minimality of the derivation) and by Lemma 10, $\mathrm{elems}(s) \subseteq E_{m-1}$ ($m \neq 1$, otherwise this step would be executed in the algorithm). As $s \in S$, then $\mathrm{elems}(s) \subseteq \mathrm{QSub}(s) \subseteq \mathrm{QSub}(E \cup \{t\})$. If $\mathrm{elems}(s) \subseteq D$ then we got contradiction with with the fact, that this step would be executed in the algorithm. If there exists $e \in \mathrm{elems}(s)$ and $e \notin D$ (that means, $e \in S$), then we get a contradiction with the minimality of m, as $e \in S$ was deduced before.

- any other composition rule, then by Lemma 8, $l_{m-1} \subseteq \mathrm{QSub}(s)$, and thus, $l_{m-1} \subseteq D \cup S$. Similarly to the previous case, $m \neq 1$ and we get a contradiction with either minimality of m, or with the fact, that the algorithm would have to add s into D.

Note, that this also shows, that decomposition rule is present in derivation.

Show, $l_k \nsubseteq D$. Suppose the opposite. Then by Lemma 8, we have $r_k \subseteq D$ what contradicts to $E_{k+1} \setminus E_k \neq \emptyset$. Thus, at least one element from l_k is not from D. Let us consider all possible decomposition rules $l_k \to r_k$:

- $\{\mathrm{pair}\,(t_1, t_2)\} \to \ulcorner t_1 \urcorner$. We know, that $\mathrm{pair}\,(t_1, t_2)$ is not in D, thus, it was built by composition. As E_i are normalized, the only possible way to build by composition $\mathrm{pair}\,(t_1, t_2)$ from normalized terms is $\{t_1, t_2\} \to \mathrm{pair}\,(t_1, t_2)$ (other ways, like $\mathrm{pair}\,(t_1, t_2), \mathrm{pair}\,(t_1, t_2) \to \ulcorner \cdot (\{\mathrm{pair}\,(t_1, t_2), \mathrm{pair}\,(t_1, t_2)\}) \urcorner$ would contradict the minimality of the derivation). Thus, t_1 was derived before (or was in D), i.e. $t_1 \in E_k$. That contradicts to $E_{k+1} \setminus E_k \neq \emptyset$.
- $\{\mathrm{pair}\,(t_1, t_2)\} \to \ulcorner t_2 \urcorner$. Similar case.
- $\{\mathrm{enc}\,(t_1, t_2), \ulcorner t_2 \urcorner\} \to \ulcorner t_1 \urcorner$. The case where $\mathrm{enc}\,(t_1, t_2) \notin D$ has similar explanations as two cases above. Thus, $\mathrm{enc}\,(t_1, t_2) \in D$. That means, $t_2 \in \mathrm{QSub}\,(E \cup \{t\})$ and $t_2 \notin D$, i.e. $t_2 \in S$. This means, t_2 was derived before and $t_2 \in S$, what contradicts to $k \leq m$.
- $\{\mathrm{aenc}\,(t_1, t_2), \ulcorner \mathrm{priv}\,(t_2) \urcorner\} \to \ulcorner t_1 \urcorner$ is a similar case to previous one. Note, that if $\mathrm{priv}\,(t_2)$ is not in D, that it must be obtained by decomposition.
- $t \to \ulcorner s \urcorner$, where $s \in \mathrm{elems}\,(t)$ and $t = \cdot\,(L)$. By Lemma 10, $\mathrm{elems}\,(t) \subseteq E_k$, that contradicts minimality of derivation ($E_{k+1} \setminus E_k \neq \emptyset$).

□

3.3 Existence of conservative solutions

In this subsection we will show that for any satisfiable constraint system, there exist a solution in special form (so called conservative solution). Roughly speaking, a solution in this form can be defined per each variable by set of quasi-subterms of the constraint system and set of atoms (also from the constraint system) that must be prived. This will bound a search space for the solution (see § 3.4).

First, we show, that on quasi-subterms of constraint system instantiated with its solution, the transformation $\pi(H\,(\cdot))$ will be a homomorphism modulo normalization.

Proposition 5. *Given a normalized constraint system $\mathcal{S}$ and its normalized solution σ. For all $t \in \mathrm{QSub}\,(\mathcal{S})$, $\ulcorner t\,\pi(\mathrm{H}\,(\sigma)) \urcorner = \ulcorner \pi(H\,(t\sigma)) \urcorner$.*

Proof. We will prove it by induction on $|\mathrm{Sub}\,(t)|$, where t is normalized.

- Let $|\mathrm{Sub}\,(t)| = 1$. Then:
  - either $t \in \mathcal{A}$. In this case $t \in (\mathcal{A} \cap \mathrm{QSub}\,(\mathcal{S}))$, 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 solution and $t \in \mathrm{QSub}\,(\mathcal{S})$, 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 $t = \mathrm{bin}\,(p,q) \in \mathrm{QSub}\,(\mathcal{S}) \Rightarrow p \in \mathrm{QSub}\,(\mathcal{S})$ and $q \in \mathrm{QSub}\,(\mathcal{S})$. 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 $\mathrm{bin}\,(p,q)\,\sigma \in \mathrm{QS\mathring{u}b}\,(S)\,\sigma$ (as $\mathrm{bin}\,(p,q) \notin \mathcal{X}$ and $t \in \mathrm{QSub}\,(\mathcal{S})$) we have that $\ulcorner \pi(H\,(\mathrm{bin}\,(p,q)\,\sigma))\urcorner = \ulcorner \pi(H\,(\mathrm{bin}\,(p\sigma,q\sigma)))\urcorner = \ulcorner \pi(H\,(\ulcorner \mathrm{bin}\,(p\sigma,q\sigma)\urcorner))\urcorner = \ulcorner \pi(H\,(\mathrm{bin}\,(\ulcorner p\sigma\urcorner,\ulcorner q\sigma\urcorner)))\urcorner = \ulcorner \pi(\{\mathrm{bin}\,(\pi(H\,(\ulcorner p\sigma\urcorner)),\pi(H\,(\ulcorner q\sigma\urcorner)))\})\urcorner =$
    $= \ulcorner \pi(\{\mathrm{bin}\,(\ulcorner \pi(H\,(p\sigma))\urcorner,\ulcorner \pi(H\,(q\sigma))\urcorner)\})\urcorner = \ulcorner \mathrm{bin}\,(\ulcorner \pi(H\,(p\sigma))\urcorner,\ulcorner \pi(H\,(q\sigma))\urcorner)\urcorner = \ulcorner \mathrm{bin}\,(\ulcorner p\,\pi(\mathrm{H}\,(\sigma))\urcorner,\ulcorner q\,\pi(\mathrm{H}\,(\sigma))\urcorner)\urcorner = \ulcorner \mathrm{bin}\,(p\,\pi(\mathrm{H}\,(\sigma)),q\,\pi(\mathrm{H}\,(\sigma)))\urcorner =$
    $= \ulcorner \mathrm{bin}\,(p,q)\,\pi(\mathrm{H}\,(\sigma))\urcorner = \ulcorner t\,\pi(\mathrm{H}\,(\sigma))\urcorner$.
  - or $t = \cdot\,(t_1,\ldots,t_m)$. As t is normalized, it implies that for all $i \in \{1,\ldots,m\}$, t_i are not in form of $\cdot\,(L_i)$ and then $t_i \in \mathrm{QSub}\,(\mathcal{S})$, and thus, we have $t_i \in \mathrm{QSub}\,(\mathcal{S}) \wedge \ulcorner \pi(H\,(t_i\sigma))\urcorner = \ulcorner t_i\,\pi(\mathrm{H}\,(\sigma))\urcorner$. $\pi(H\,(t\sigma)) = \pi(H\,(\cdot\,(t_1\sigma,\ldots,t_m\sigma))) = \pi(H\,(t_1\sigma) \cup \cdots \cup H\,(t_m\sigma)) =$ (by Statement 10 of Lemma 3) $= \pi(\{\pi(H\,(t_1\sigma)),\ldots,\pi(H\,(t_m\sigma))\}) = \pi(\{\ulcorner t_1\,\pi(\mathrm{H}\,(\sigma))\urcorner,\ldots,\ulcorner t_m\,\pi(\mathrm{H}\,(\sigma))\urcorner\}) = \ulcorner \cdot\,(\ulcorner t_1\,\pi(\mathrm{H}\,(\sigma))\urcorner,\ldots,\ulcorner t_m\,\pi(\mathrm{H}\,(\sigma))\urcorner)\urcorner = \ulcorner \cdot\,(t_1\,\pi(\mathrm{H}\,(\sigma)),\ldots,t_m\,\pi(\mathrm{H}\,(\sigma)))\urcorner = \ulcorner (\cdot\,(t_1,\ldots,t_m))\,\pi(\mathrm{H}\,(\sigma))\urcorner = \ulcorner t\,\pi(\mathrm{H}\,(\sigma))\urcorner$
  - or $t = \mathrm{priv}\,(q)$. Then $q \in \mathrm{QSub}\,(\mathcal{S})$.
    $\pi(H\,(t\sigma)) = \pi(\{\mathrm{priv}\,(\pi(H\,(q\sigma)))\}) = \ulcorner \mathrm{priv}\,(\pi(H\,(q\sigma)))\urcorner = \ulcorner \mathrm{priv}\,(q\,\pi(\mathrm{H}\,(\sigma)))\urcorner = \ulcorner \mathrm{priv}\,(q)\,\pi(\mathrm{H}\,(\sigma))\urcorner = \ulcorner t\,\pi(\mathrm{H}\,(\sigma))\urcorner$.

Thus, the proposition is proven.

□

Now we show, that relation of derivability between a term and a set of terms is stable with regard to transformation $\pi(H\,(\cdot))$.

Lemma 11. *Given a normalized constraint system $\mathcal{S}$ and its normalized solution σ. For any DY+ACI rule $l_1,\ldots,l_k \to r$, $\pi(H\,(r)) \in \mathrm{Der}\,(\{\pi(H\,(l_1)),\ldots,\pi(H\,(l_k))\})$.*

. 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:

  - $\exists u \in \mathrm{QS\mathring{u}b}\,(S)$ such that $\ulcorner \mathrm{pair}\,(t_1,t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H\,(\ulcorner \mathrm{pair}\,(t_1,t_2)\urcorner)) = \pi(H\,(\mathrm{pair}\,(\ulcorner t_1\urcorner,\ulcorner t_2\urcorner))) = \pi(\{\mathrm{pair}\,(\pi(H\,(\ulcorner t_1\urcorner)),\pi(H\,(\ulcorner t_2\urcorner)))\}) =$
    $= \ulcorner \mathrm{pair}\,(\pi(H\,(t_1)),\pi(H\,(t_2)))\urcorner$ and then
    $\pi(H\,(\ulcorner \mathrm{pair}\,(t_1,t_2)\urcorner)) \in \mathrm{Der}\,(\{\pi(H\,(t_1)),\pi(H\,(t_2))\})$.

- $\nexists u \in \mathring{\mathrm{QSub}}(S)$ such that $\ulcorner \mathrm{pair}(t_1, t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then (by definition, Lemma 3 and Proposition 1)
  $\pi(H(\ulcorner \mathrm{pair}(t_1, t_2)\urcorner)) = \pi(H(\ulcorner t_1\urcorner) \cup H(\ulcorner t_2\urcorner)) \in \mathrm{Der}(\ulcorner H(t_1) \cup H(t_2)\urcorner)$. By Proposition 1, $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(t_1))\})$ and $\ulcorner H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(t_2))\})$, then by Lemma 2, $\ulcorner H(t_1)\urcorner \cup \ulcorner H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(t_1))\} \cup \{\pi(H(t_2))\})$. Now, by applying Lemma 1, we have that
  $\pi(H(\ulcorner \mathrm{pair}(t_1, t_2)\urcorner)) \in \mathrm{Der}(\{\pi(H(t_1))\} \cup \{\pi(H(t_2))\})$.

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.
- $t_1, \mathrm{priv}(t_2) \to \ulcorner \mathrm{sig}(t_1, \mathrm{priv}(t_2))\urcorner$.
  - $\exists u \in \mathring{\mathrm{QSub}}(S)$ 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 $\pi(H(\mathrm{priv}(t_2))) = \mathrm{priv}(\pi(H(t_2))))$.
  - $\nexists u \in \mathring{\mathrm{QSub}}(S)$ 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 2,
  $\pi(H(\ulcorner \cdot(t_1, \ldots, t_m)\urcorner)) \in \mathrm{Der}(\{\pi(H(t_1)), \ldots, \pi(H(t_m))\})$.
- $\mathrm{enc}(t_1, t_2), \ulcorner t_2\urcorner \to \ulcorner t_1\urcorner$. 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:
  - $\exists u \in \mathring{\mathrm{QSub}}(S)$ 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 then $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in$
    $\mathrm{Der}(\{\mathrm{enc}(\pi(H(t_1)), \pi(H(t_2))), \ulcorner \pi(H(\ulcorner t_2\urcorner))\urcorner\})$.
  - $\nexists u \in \mathring{\mathrm{QSub}}(S)$ such that $\ulcorner \mathrm{enc}(t_1, t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{enc}(t_1, t_2))) = \pi(H(t_1) \cup H(t_2))$. Using Proposition 1, we have
    $\ulcorner H(t_1) \cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{enc}(t_1, t_2)))\})$. Thus (by Lemma 3) $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{enc}(t_1, t_2)))\})$. And then, by Proposition 1 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 1, 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:

- $\exists u \in \mathring{\mathrm{QSub}}(S)$ such that $\ulcorner \mathrm{aenc}(t_1, t_2)\urcorner = \ulcorner u\sigma\urcorner$. Then $\pi(H(\mathrm{aenc}(t_1, t_2))) = \mathrm{aenc}(\pi(H(t_1)), \pi(H(t_2)))$ 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, $\pi(H(\ulcorner \mathrm{priv}(t_2)\urcorner)) = \pi(H(\mathrm{priv}(t_2))) = \pi(\{\mathrm{priv}(\pi(H(t_2)))\}) = \ulcorner \mathrm{priv}(\pi(H(t_2)))\urcorner$.
- $\nexists u \in \mathrm{QSub}(S)$ 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 Proposition 1, we have $\ulcorner H(t_1) \cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{aenc}(t_1, t_2)))\})$, thus (by Lemma 3) $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{aenc}(t_1, t_2)))\})$. And then, by Proposition 1 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 1, we have $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\pi(H(\mathrm{aenc}(t_1, t_2))))$.

- $\mathrm{pair}(t_1, t_2) \to \ulcorner t_1\urcorner$. Here, as usual, we consider two cases:
  - $\exists u \in \mathring{\mathrm{QSub}}(S)$ 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 $\pi(H(\ulcorner t_1\urcorner)) = \ulcorner \pi(H(t_1))\urcorner \in \mathrm{Der}(\{\pi(H(\mathrm{pair}(t_1, t_2)))\})$.
  - $\nexists u \in \mathring{\mathrm{QSub}}(S)$ 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 Proposition 1, we have $\ulcorner H(t_1) \cup H(t_2)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{pair}(t_1, t_2)))\})$, thus $\ulcorner H(t_1)\urcorner \subseteq \mathrm{Der}(\{\pi(H(\mathrm{pair}(t_1, t_2)))\})$. And then, by Proposition 1 we have that $\pi(H(t_1)) \in \mathrm{Der}(\ulcorner H(t_1)\urcorner)$. Therefore, by Lemma 1, we have $\pi(H(\ulcorner t_1\urcorner)) = \pi(H(t_1)) \in \mathrm{Der}(\pi(H(\mathrm{pair}(t_1, t_2))))$.

- $\mathrm{pair}(t_1, t_2) \to \ulcorner t_2\urcorner$. 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 Proposition 1, $\ulcorner H(t_1) \cup \cdots \cup H(t_m)\urcorner \subseteq \mathrm{Der}(\pi(H(\cdot(t_1, \ldots, t_m))))$; and then $\ulcorner H(t_i)\urcorner \subseteq \mathrm{Der}(\pi(H(\cdot(t_1, \ldots, t_m))))$. As $\pi(H(t_i)) \in \mathrm{Der}(\ulcorner H(t_i)\urcorner)$, by Lemma 1 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. □

Using Proposition 5 and Lemma 11 we will show, that transformation $\pi(H(\cdot))$ preserves the property of substitution to be a solution.

Theorem 1. *Given a normalized constraint system $\mathcal{S}$ and its normalized solution σ. Then substitution $\pi(\mathrm{H}(\sigma))$ also satisfies $\mathcal{S}$.*

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

By Lemma 11 and Lemma 2 we can easily prove that if $k > 0$, then $\pi(H(A_j)) \subseteq \mathrm{Der}(\pi(H(A_{j-1})))$, $j = 1, \ldots, k$. Then, applying transitivity of $\mathrm{Der}(\cdot)$ (Lemma 1) 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 5 we have that $\pi(H(A_0)) = \pi(H(E\sigma)) = \ulcorner E\,\pi(\mathrm{H}(\sigma))\urcorner$, as $E \subseteq \mathrm{QSub}(\mathcal{S})$. 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 $\ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner \in \pi(H(A_k))$. Thus, we have $\ulcorner t\,\pi(\mathrm{H}(\sigma))\urcorner \in \pi(H(A_k)) \subseteq \mathrm{Der}(\pi(H(A_0))) = \mathrm{Der}(\ulcorner E\,\pi(\mathrm{H}(\sigma))\urcorner)$, that means $\pi(\mathrm{H}(\sigma))$ satisfies any constraint of $\mathcal{S}$.

□

From now till the end of subsection we will study a very useful property of $\pi(\mathrm{H}(\sigma))$. Proposition 6 and its corollary show, that if constraint system has a normalized solution (σ) which sends different variables to different values, then there exists another normalized solution $(\pi(\mathrm{H}(\sigma)))$ that sends any variable of its domain to an ACI-set of some non-variable quasi-subterms of constraint system instantiated by itself and some private keys built with atoms of the constraint system.

Lemma 12. *If* $\ulcorner u\sigma\urcorner = \mathrm{enc}(p,q)$, σ *is normalized,* $u = \ulcorner u\urcorner$ *and* $u \notin \mathcal{X}$ *and* $x\sigma \neq y\sigma, x \neq y$ *then* $\exists s \in \mathrm{QSub}(u)$ *such that* $s = \mathrm{enc}(p',q')$ *and* $\ulcorner s\sigma\urcorner = \mathrm{enc}(p,q)$. *The similar is true in the case of* $\ulcorner u\sigma\urcorner = \mathrm{pair}(p,q)$, $\ulcorner u\sigma\urcorner = \mathrm{aenc}(p,q)$, $\ulcorner u\sigma\urcorner = \mathrm{sig}(p,q)$ *and for* $\ulcorner u\sigma\urcorner = \mathrm{priv}(p)$.

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

- u not in form of $\cdot(L)$. Then, as $u \notin \mathcal{X}$ and $\ulcorner u\sigma\urcorner = \mathrm{enc}(p,q)$, we have $u = \mathrm{enc}(p',q')$ (where $\ulcorner p'\sigma\urcorner = p$ and $\ulcorner q'\sigma\urcorner = q$). Then we can choose $s = \mathrm{enc}(p',q') = u \in \mathrm{QSub}(u)$.
- $u = \cdot(t_1,\ldots,t_m)$, $m > 1$, as $u = \ulcorner u\urcorner$. Then, for all i, t_i is either a variable, or $\mathrm{enc}(p'_i,q'_i)$. But, as $x\sigma \neq y\sigma, x \neq y$ and as σ is normalized, we can claim, that $\{t_1,\ldots,t_m\}$ contains at most one variable. Then, as $m > 1$, there exists i such that $t_i = \mathrm{enc}(p'_i,q'_i)$. Then by definition of normalization function, and from $\ulcorner u\sigma\urcorner = \mathrm{enc}(p,q)$ we have, that $\ulcorner \mathrm{elems}(u\sigma)\urcorner = \{\mathrm{enc}(p,q)\}$ and as $t_i\sigma$ is an element of $u\sigma$, we have $\ulcorner \mathrm{enc}(p'_i,q'_i)\,\sigma\urcorner = \mathrm{enc}(p,q)$. Thus, we can choose $s = t_i$, as $t_i \in \mathrm{QSub}(u)$ and $t_i = \mathrm{enc}(p'_i,q'_i)$.

The other cases (pair, priv, etc...) can be proved similarly.

□

Proposition 6. *Given a normalized constraint system* $\mathcal{S}$ *and its normalized solution* σ *such that* $\forall x,y \in \mathrm{dom}(\sigma)$, $x \neq y \implies x\sigma \neq y\sigma$. *Then* $\forall x \in \mathrm{dom}(\pi(\mathrm{H}(\sigma)))$ $\exists k \in \mathbb{N}$ $\exists s_1,\ldots,s_k \in \mathrm{Q\mathring{S}ub}(S) \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$ *such that* $\mathrm{root}(s_i) \neq \cdot$ *and* $x\,\pi(\mathrm{H}(\sigma)) = \pi(\{s_1\,\pi(\mathrm{H}(\sigma)),\ldots,s_k\,\pi(\mathrm{H}(\sigma))\})$.

Proof. 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 $H(\cdot)$, $s \in (\mathcal{A} \cap \mathrm{QSub}(\mathcal{S}))$. Thus, $s\,\pi(\mathrm{H}(\sigma)) = s$, $s \in \mathrm{Q\mathring{S}ub}(S)$, $s \neq \cdot(L)$;

- or $s = \mathrm{bin}\,(\pi(H\,(t_1)), \pi(H\,(t_2)))$ and $\exists u \in \mathrm{QS\mathring{u}b}\,(S)$ 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 12 are satisfied, $\exists v \in \mathrm{QSub}\,(u)$ such that $\ulcorner v\sigma \urcorner = \mathrm{bin}\,(\ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner)$ and $v = \mathrm{bin}\,(p, q)$ and as $u \in \mathrm{QS\mathring{u}b}\,(S)$ then $v \in \mathrm{QS\mathring{u}b}\,(S)$. By Proposition 5, $\ulcorner v\,\pi(\mathrm{H}\,(\sigma)) \urcorner = \pi(H\,(v\sigma)) = \pi(H\,(\ulcorner v\sigma \urcorner)) = \pi(H\,(\mathrm{bin}\,(t_1, t_2))) = \pi(\{\mathrm{bin}\,(\pi(H\,(t_1)), \pi(H\,(t_2)))\}) = \mathrm{bin}\,(\pi(H\,(t_1)), \pi(H\,(t_2))) = s$. That means, $\exists v \in \mathrm{QS\mathring{u}b}\,(S)\,, v \neq \cdot\,(L)$ 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, moreover, by definition of $H\,(\cdot)$, this atom is from $(\mathcal{A} \cap \mathrm{QSub}\,(\mathcal{S}))$. Therefore, $s = \mathrm{priv}\,(a)$, where $a \in \mathcal{A} \cap \mathrm{QSub}\,(\mathcal{S})$ (and of course, $s \neq \cdot\,(L)$).

Thus, $\forall s \in H\,(x\sigma)$, $\exists v \in (\mathrm{QSub}\,(\mathcal{S})) \cup \mathrm{priv}\,(\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A}) \setminus \mathcal{X} \mid s = \ulcorner v\,\pi(\mathrm{H}\,(\sigma)) \urcorner$. 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 $s_1, \ldots, s_k \in \mathrm{QS\mathring{u}b}\,(\mathcal{S}) \cup \mathrm{priv}\,(\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A})$ and $s_i \neq \cdot\,(L)\,, \forall 1 \leq i \leq k$. That proves the proposition. □

Corollary 1. *Given normalized constraint system $\mathcal{S}$ and its normalized solution σ' such that $x \neq y \implies x\sigma' \neq y\sigma'$. Then there exists a normalized solution σ of $\mathcal{S}$ such that $\forall x \in \mathrm{dom}\,(\sigma)$ $\exists k \in \mathbb{N}\, \exists s_1, \ldots, s_k \in \mathrm{QS\mathring{u}b}\,(S) \cup \mathrm{priv}\,(\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A})\,,\ x\sigma = \pi(\{s_1\sigma, \ldots, s_k\sigma\})$ and $s_i \neq s_j$, if $i \neq j$; $s_i \neq \cdot\,(L)\,, \forall i$.*

Any normalized solution with property shown in Corollary 1 we will call *conservative*.

3.4 Bounds on conservative solutions

To get a decidability result, we first show an upper bound on size of conservative solution and then, by reducing any satisfiable constraint system to one that have conservative solution and showing that reduced one is smaller (by size) than original one, we obtain an existence of a solution with bounded size for any satisfiable constraint system.

Lemma 13. *Given a normalized constraint system $\mathcal{S}$ and its conservative solution σ. Then $\forall x \in \mathrm{Vars}\,(\mathcal{S})\,, \mathrm{QSub}\,(x\sigma) \subseteq \ulcorner \mathrm{QSub}\,(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}\,(\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A})$.*

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

By Proposition 6 $\forall x\ x\sigma = \pi(\{s_1^x\sigma, \ldots, s_{k^x}^x\sigma\})$, where $s_i^x \in (\mathrm{QSub}\,(\mathcal{S}) \setminus \mathcal{X}) \cup \mathrm{priv}\,(\mathrm{QSub}\,(\mathcal{S}) \cap \mathcal{A})$ and $s_i^x \neq \cdot\,(L)$.

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 $\mathrm{size}_{\mathrm{DAG}}\,(x\sigma) = \mathrm{size}_{\mathrm{DAG}}\,(\pi(\{s_1^x\sigma, \ldots, s_{k^x}^x\sigma\})) = \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner \cdot\,(s_1^x\sigma, \ldots, s_{k^x}^x\sigma) \urcorner) \geq$ (by Lemma 3) $\geq \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner s_i^x\sigma \urcorner) > \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner y\sigma \urcorner)$, 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 first case, $\mathrm{size}_{\mathrm{DAG}}\,(\ulcorner s_i^x\sigma \urcorner) =$

$= \mathrm{size}_{\mathrm{DAG}}(\mathrm{bin}(\ulcorner p\sigma \urcorner, \ulcorner q\sigma \urcorner)) = 1 + \mathrm{size}_{\mathrm{DAG}}(\{\ulcorner p\sigma \urcorner, \ulcorner q\sigma \urcorner\})$ and as $y \in \mathrm{Vars}(\{p, q\})$, using Statement 18 of Lemma 3, we get $\mathrm{size}_{\mathrm{DAG}}(\ulcorner s_i^x \sigma \urcorner) \geq 1 + \mathrm{size}_{\mathrm{DAG}}(\ulcorner y\sigma \urcorner))$ And as $\mathrm{size}_{\mathrm{DAG}}(\ulcorner y\sigma \urcorner) = \mathrm{size}_{\mathrm{DAG}}(y\sigma)$ That means, $y \sqsubset x$. 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 $x\sigma = \ulcorner \cdot (s_1^x, \ldots, s_{k^x}^x) \urcorner$. We have that $\mathrm{QSub}(x\sigma) = \{\ulcorner \cdot (s_1^x, \ldots, s_{k^x}^x) \urcorner\} \cup \mathrm{QSub}(s_1^x) \cup \cdots \cup \mathrm{QSub}(s_{k^x}^x) \subseteq \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathcal{A} \cap \mathrm{QSub}(\mathcal{S}))$, as $\forall s \in \mathrm{QSub}(s_i^x)$, $s \in \mathcal{T}_g$ and $s \in \mathrm{QSub}(\mathcal{S})$ or $s = \mathrm{priv}(a)$ or $s = a$, where $a \in \mathrm{QSub}(\mathcal{S}) \cap \mathcal{A}$, therefore $s = \ulcorner s \urcorner = s\sigma \in \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$ and $\ulcorner \cdot (s_1^x, \ldots, s_{k^x}^x) \urcorner = x\sigma \in \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner$.
- Suppose, for all $z \sqsubset y$, $\mathrm{QSub}(z\sigma) \subseteq \ulcorner \mathrm{QSub}(\mathcal{S}\sigma) \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$.
- Show, that $\mathrm{QSub}(y\sigma) \subseteq \mathrm{QSub}(\mathcal{S}\sigma) \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$. 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 $\mathrm{QSub}(y\sigma) = \{y\sigma\} \cup \mathrm{QSub}(\ulcorner s_1^y\sigma \urcorner) \cup \cdots \cup \mathrm{QSub}(\ulcorner s_{k^y}^y\sigma \urcorner)$. We know that $y\sigma \in \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner$. Let us show that $\mathrm{QSub}(\ulcorner s_i^y\sigma \urcorner) \subseteq \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$. By Lemma 3 we have $\mathrm{QSub}(\ulcorner s_i^y\sigma \urcorner) \subseteq \ulcorner \mathrm{QSub}(s_i^y\sigma) \urcorner \subseteq$
$\subseteq \ulcorner \mathrm{QSub}(s_i^y)\,\sigma \cup \mathrm{QSub}(\mathrm{Vars}(s_i^y)\,\sigma) \urcorner = \ulcorner \mathrm{QSub}(s_i^y)\,\sigma \urcorner \cup \mathrm{QSub}(\mathrm{Vars}(s_i^y)\,\sigma)$. We can see that $\ulcorner \mathrm{QSub}(s_i^y)\,\sigma \urcorner \subseteq \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$ (as $s_i^y \in \mathrm{QSub}(\mathcal{S}) \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$); and by induction supposition and by statement proved above we have $\mathrm{QSub}(\mathrm{Vars}(s_i^y)\,\sigma) \subseteq \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$. Thus, $\mathrm{QSub}(y\sigma) \subseteq \ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$.

□

Proposition 7. *For normalized constraint system $\mathcal{S}$ that have conservative solution σ, $\forall x \in \mathrm{Vars}(\mathcal{S}), \mathrm{size}_{\mathrm{DAG}}(x\sigma) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$.*

Proof. As $|\ulcorner \mathrm{Sub}(\mathcal{S})\,\sigma \urcorner| \leq |\mathrm{Sub}(\mathcal{S})\,\sigma| \leq |\mathrm{Sub}(\mathcal{S})| = \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$, we have (using the fact that σ is normalized and Lemma 13) that $|\mathrm{Sub}(x\sigma)| = |\mathrm{QSub}(x\sigma)| \leq |\ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner \cup \mathrm{priv}(\mathcal{A} \cap \mathrm{QSub}(\mathcal{S}))| \leq |\ulcorner \mathrm{QSub}(\mathcal{S})\,\sigma \urcorner| + |\mathrm{priv}(\mathcal{A} \cap \mathrm{QSub}(\mathcal{S}))| \leq \mathrm{size}_{\mathrm{DAG}}(\mathcal{S}) +$
$+ |\mathcal{A} \cap \mathrm{QSub}(\mathcal{S})| \leq 2 \times \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$; thus, $\mathrm{size}_{\mathrm{DAG}}(x\sigma) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$. □

From this proposition and Corollary 1 we obtain an existence of bounded solution for a normalized constraint system that have a solution sending different variables to different values. We will reduce an arbitrary constraint system to already studied case. The target properties are stated in Proposition 8 and Corollary 2.

Lemma 14. *Given any constraint system $\mathcal{S}$ and any substitution θ such that* $\mathrm{dom}(\theta) = \mathrm{Vars}(\mathcal{S})$ *and* $\mathrm{dom}(\theta)\,\theta \subseteq \mathrm{dom}(\theta)$. *Then* $\mathrm{size}_{\mathrm{DAG}}(\mathcal{S}\theta) \leq \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$.

Proof. By Lemma 3, $\mathrm{size}_{\mathrm{DAG}}(\mathcal{S}\theta) = |\mathrm{Sub}(\mathcal{S}\theta)| = |\mathrm{Sub}(\mathcal{S})\,\theta \cup \mathrm{Sub}(\mathrm{Vars}(\mathcal{S})\,\theta)|$, but $\mathrm{Vars}(\mathcal{S})\,\theta \subseteq \mathrm{dom}(\theta) = \mathrm{Vars}(\mathcal{S})$ ($\mathrm{Vars}(\mathcal{S}\sigma)$ 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,
$\mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S}\theta) = |\mathrm{Sub}\,(\mathcal{S})\,\theta| \leq |\mathrm{Sub}\,(\mathcal{S})| = \mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S})$. □

Definition 3.17. *Let σ and δ be substitutions. Then $\sigma[\delta]$ is a substitution such that* $\mathrm{dom}\,(\sigma[\delta]) = \mathrm{dom}\,(\delta)$ *and* $\forall x \in \mathrm{dom}\,(\sigma[\delta])\,,\ x\sigma[\delta] = (x\delta)\sigma$.

Lemma 15. *Let θ and σ be substitutions such that* $\mathrm{dom}\,(\theta)\,\theta = \mathrm{dom}\,(\sigma)$, $\mathrm{dom}\,(\sigma) \subseteq \mathrm{dom}\,(\theta)$ *and σ is ground. Then, for any term t,* $(t\theta)\sigma = t\sigma[\theta]$.

Proof. When 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 to $(x\theta)\sigma$ (as $\mathrm{dom}\,(\theta)\,\theta = \mathrm{dom}\,(\sigma)$); and no other variables will be replaced (as $\mathrm{dom}\,(\sigma) \subseteq \mathrm{dom}\,(\theta)$). Thus, we can see that it is the same as in definition of $\sigma[\theta]$. □

Proposition 8. *Given any satisfiable constraint system $\mathcal{S}$. Then there exists a solution σ of $\mathcal{S}$ such that* $\forall x \in \mathrm{dom}\,(\sigma)\,,\ \mathrm{size}_{\mathrm{DAG}}\,(x\sigma) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner\mathcal{S}\urcorner)$

Proof. From proposition 2 and 3 we know that if σ' is a solution of $\mathcal{S}$ then $\ulcorner\sigma'\urcorner$ is a solution of $\mathcal{S}$ and $\ulcorner\sigma'\urcorner$ is a solution of $\ulcorner\mathcal{S}\urcorner$. Then, there exists a substitution $\theta : \mathrm{dom}\,(\theta) = \mathrm{dom}\,(\ulcorner\sigma'\urcorner)$, $\mathrm{dom}\,(\theta)\,\theta \subseteq \mathrm{dom}\,(\theta)\,,\ \sigma'' = \ulcorner\sigma'\urcorner|_{\mathrm{dom}(\theta)\theta}$ and σ'' is a solution 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 solution of $\ulcorner\mathcal{S}\urcorner\theta$.

Then, as σ'' is a solution of $\ulcorner\mathcal{S}\urcorner\theta$, using Proposition 2, we can say, that σ'' is a solution 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 1, which gives us existence of conservative solution δ of $\ulcorner\ulcorner\mathcal{S}\urcorner\theta\urcorner$. That is why we can apply Proposition 7: $\forall x \in \mathrm{Vars}\,(\ulcorner\ulcorner\mathcal{S}\urcorner\theta\urcorner)\,, \mathrm{size}_{\mathrm{DAG}}\,(x\delta) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner\ulcorner\mathcal{S}\urcorner\theta\urcorner)$.

Note, that using Proposition 2, Lemma 15 and definition of "solution", we can easily show that $\delta[\theta]$ is a solution 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 $\mathrm{size}_{\mathrm{DAG}}\,(x\delta[\theta]) = \mathrm{size}_{\mathrm{DAG}}\,(y\delta) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner\ulcorner\mathcal{S}\urcorner\theta\urcorner) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner\mathcal{S}\urcorner\theta)$. Applying Lemma 14, we have $\mathrm{size}_{\mathrm{DAG}}\,(x\delta[\theta]) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner\mathcal{S}\urcorner)$.

Summing up, we have a normalized solution $\sigma = \delta[\theta]$ of $\ulcorner\mathcal{S}\urcorner$ such that $\forall x \in \mathrm{dom}\,(\sigma)$, $\mathrm{size}_{\mathrm{DAG}}\,(x\sigma) \leq 2 \times \mathrm{size}_{\mathrm{DAG}}\,(\ulcorner\mathcal{S}\urcorner)$. □

Corollary 2. *Constraint system $\mathcal{S}$ is satisfiable if and only if there exists a normalized solution of $\mathcal{S}$ defined on* $\mathrm{Vars}\,(\mathcal{S})$ *which maps a variable to a ground term in* $\mathcal{T}(\mathcal{A} \cap \mathrm{QSub}\,(\ulcorner\mathcal{S}\urcorner)\,, \emptyset)$ *with size not greater than double* $\mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S})$.

Using this result, we propose an algorithm of satisfiability of constraint system (Algorithm 3).

Algorithm 3: Solving constraint system

Input: A constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$
Output: Model σ, if exists; otherwise $\perp$

```
Guess for every variable of S a value of ground substitution σ with size not greater
  than 2 × size_DAG (S);
if σ satisfies E_i ▷ t_i for all i = 1, ..., n then
    return σ
else
    return ⊥
end
```

Proposition 9. *Algorithm 3 is correct.*

Proof. Let σ be an output of Algorithm 3. 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 solution of $\mathcal{S}$. □

Proposition 10. *Algorithm 3 is complete.*

Proof. Suppose, $\mathcal{S}$ is satisfiable. Then, by Corollary 2, there exists a guess of value of ground substitution on every element of $\mathrm{Vars}(\mathcal{S})$ with size not greater than $2 \times \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$ which represents a solution σ of $\mathcal{S}$. Thus, algorithm 3 will return this σ.

□

4 Complexity analysis

In this section we present complexity classes of proposed algorithms. First, we expose what we use as a representation of constraint systems to justify the selected measure of algorithms inputs. Then, we notice that normalization algorithm is polynomial in time. After that we will show the polynomial complexity of the ground derivability algorithm. And as a consequence of the results given before, we obtain that the proposed algorithm for solving general constraint system within DY+ACI model is in NP.

To reason about complexity, we have to define a size of its input. For terms and set of terms, we will use $\mathrm{size}_{\mathrm{DAG}}(\cdot)$. For system of constraints $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ we will use $n \times \mathrm{size}_{\mathrm{DAG}}(\mathcal{S})$. The justification is given below.

Definition 4.1. DAG-representation *of a constraint system* $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ *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* $2n$*-bit integer, where* $m[2i-1] = 1 \iff f(v) \in E_i$ *and* $m[2i] = 1 \iff f(v) = t_i$.
- $v_1 \xrightarrow{1} v_2 \in \mathbb{E} \iff \exists p \in \mathcal{T} : (\exists\,\mathrm{bin} : f(v_1) = \mathrm{bin}\,(f(v_2), p)) \vee f(v_1) = \mathrm{priv}\,(f(v_2))$;
- $v_1 \xrightarrow{2} v_2 \in \mathbb{E} \iff \exists p \in \mathcal{T} : \exists\,\mathrm{bin} : f(v_1) = \mathrm{bin}\,(p, f(v_2))$;
- $v_1 \xrightarrow{i} v_2 \in \mathbb{E} \iff f(v_1) = \cdot\,(L) \wedge L[i] = f(v_2)$;

Example 7. *A constraint system*

$$\mathcal{S} = \left\{ \begin{array}{rl} \{\mathrm{enc}\,(a,x)\,, \mathrm{pair}\,(b, \mathrm{enc}\,(a,a))\,, c\} & \rhd a \\ \{\mathrm{priv}\,(b)\,, c\} & \rhd y \\ \{\mathrm{enc}\,(\mathrm{sig}\,(a, \mathrm{priv}\,(c))\,, y)\,, \mathrm{aenc}\,(x,b)\} & \rhd \mathrm{pair}\,(\mathrm{enc}\,(a,x)\,, c) \end{array} \right.$$

will be represented as shown[4] *in Figure 4. Nodes of this graph represent an element from* $\mathrm{Sub}\,(\mathcal{S})$ *by indicating its root symbol (first part of its tag) and pointers to the children.*

Remark, that this representation can be refined, as we know that RHS of a constraint is exactly one term. That is why we could tag a node not with $2n$ bits but with $n + \lceil \log(n+1) \rceil$ bits (concerning the second component of the tagging function).

As number of edges does not exceed a squared number of nodes, such a representation takes not more than $P(n \times \mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S}))$ bits of space, where P is some polynomial with non-negative coefficients. On the other hand, as we are not interested in rigorous estimation of complexity, but work in a polynomial class, we will measure complexity of algorithms by taking $n \times \mathrm{size}_{\mathrm{DAG}}\,(\mathcal{S})$ as size of constraint system $\mathcal{S}$.

The DAG-representation of a term t has the similar structure as it was shown for constraint system apart from the second part of a tagging function: we need only $\mathrm{root}\,(f(v))$ as a node's tag. The size of this representation will be polynomially bounded by $\mathrm{size}_{\mathrm{DAG}}\,(t)$.

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

Lemma 16. *Given a term* t*. Normalization can be done in polynomial time on* $\mathrm{size}_{\mathrm{DAG}}\,(t)$.

Proof idea. The algorithm of normalization works bottom-up by flattening nested ACI-sets, sorting children of ACI-set nodes, removing duplicates and removing nodes without incoming edges (except the root of t). □

Proposition 11. *The general constraint system within DY+ACI satisfiability problem, that Algorithm 3 solves, is in* NP.

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

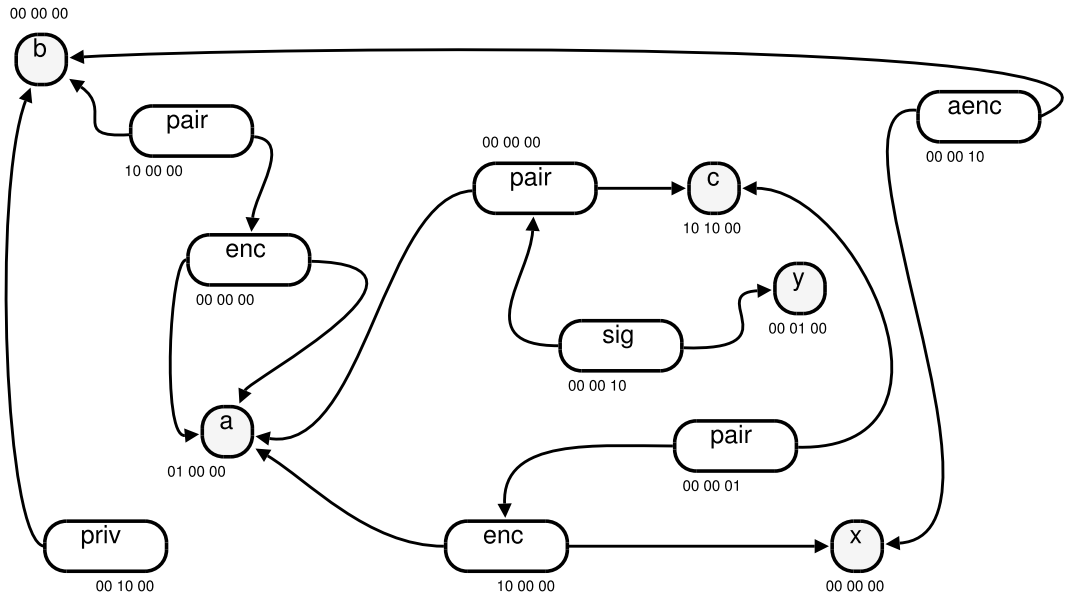

Figure 4: DAG-representation of constraint system $\mathcal{S}$

Proof. Algorithm 3 returns a proof for decision problem if it exists. We have to show, that verification of this proof takes a polynomial time with regard to initial problem size. As the size, we will consider space needed to write down input's DAG-representation.

First, we will normalize $\mathcal{S}\sigma$. From Lemma 16 follows, that we can do it for the time $T_{\ulcorner\urcorner} \leq \sum_{E \rhd t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} P_{\ulcorner\urcorner}(\text{size}_{\text{DAG}}(q\sigma))$, where $P_{\ulcorner\urcorner}$ is some polynomial with non-negative coefficients.

From the Proposition 12 we will know that check of derivability of a normalized ground term g from set of normalized ground terms G takes a polynomial time depending on $\text{size}_{\text{DAG}}(G \cup \{g\})$. I.e. there exists a polynomial P_g with non-negative coefficients, such that number of operations (execution time) to verify the derivability (g from G) will be limited by $P_g(\text{size}_{\text{DAG}}(G \cup \{g\}))$. Then the execution time for checking a set of ground constraints $\{G_i \rhd g_i\}_{i=1,\ldots,n}$ will be limited by $\sum_{i=1}^{n} P_g(\text{size}_{\text{DAG}}(G_i \cup \{g_i\}))$.

The algorithm gives a proof σ (a normalized substitution) such that $\forall x \in \text{dom}(\sigma)$, $\text{size}_{\text{DAG}}(x\sigma) \leq 2 \times \text{size}_{\text{DAG}}(\mathcal{S})$.

To show that the algorithm is in NP we need to show, that execution time of check is polynomial limited by measure of algorithm's input, i.e. there exists a polynomial P, such that execution time does not exceed $O(P(n \times \text{size}_{\text{DAG}}(\mathcal{S})))$ steps.

In our case, execution time T of a check will be $T = T_{\ulcorner\urcorner} + T_g$, where T_g is a time needed for checking ground derivability of $\mathcal{S}\sigma$: $T_g \leq \sum_{i=1}^{n} P_g(\text{size}_{\text{DAG}}(\ulcorner(E_i \cup \{t_i\})\sigma\urcorner)$. As P_g is a polynomial, let us say, of degree $m' > 0$, with non-negative coefficients (as well

as $P_{\ulcorner\urcorner}$ of degree $m'' > 0$), we can use the fact, that for any positive integers $x_1, \ldots, x_k$, $\sum_{i=1}^{k} P_g(x_i) \leq P_g(\sum_{i=1}^{k} x_i)$. Then we have $T_g \leq P_g(\sum_{i=1}^{n} \text{size}_{\text{DAG}}(\ulcorner(E_i \cup \{t_i\})\sigma\urcorner))$ and by Statement 16 of Lemma 3 we have $T_g \leq P_g(\sum_{i=1}^{n} \text{size}_{\text{DAG}}((E_i \cup \{t_i\})\sigma))$; using the same lemma, we have

$$
\begin{aligned}
T_g \leq P_g\left(\sum_{i=1}^{n}\left(\text{size}_{\text{DAG}}(E_i \cup \{t_i\}) + \text{size}_{\text{DAG}}\left(\bigcup_x x\sigma\right)\right)\right) \leq \\
\leq P_g\left(\sum_{i=1}^{n}\left(\text{size}_{\text{DAG}}(E_i) + \text{size}_{\text{DAG}}(t_i) + \sum_x \text{size}_{\text{DAG}}(x\sigma)\right)\right) \leq \\
\leq P_g\left(\sum_{i=1}^{n}(2\,\text{size}_{\text{DAG}}(\mathcal{S})) + n \times \sum_x (\text{size}_{\text{DAG}}(x\sigma))\right) \leq \\
\leq P_g\left(2 \times n \times \text{size}_{\text{DAG}}(\mathcal{S}) + n \times \sum_x (2 \times \text{size}_{\text{DAG}}(\mathcal{S}))\right) \leq \\
\leq P_g\left(2 \times n \times \text{size}_{\text{DAG}}(\mathcal{S}) + 2 \times n \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2\right) \leq \\
\leq P_g\left(4 \times n \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2\right) \leq P_g\left(4 \times (n \times \text{size}_{\text{DAG}}(\mathcal{S}))^2\right) = \\
= O\left((n \times \text{size}_{\text{DAG}}(\mathcal{S}))^{2m'}\right).
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
T_{\ulcorner\urcorner} \leq \sum_{E \triangleright t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} P_{\ulcorner\urcorner}(\text{size}_{\text{DAG}}(q\sigma)) \leq P_{\ulcorner\urcorner}\left(\sum_{E \triangleright t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} \text{size}_{\text{DAG}}(q\sigma)\right) \leq \\
\leq P_{\ulcorner\urcorner}\left(\sum_{E \triangleright t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} \left(\text{size}_{\text{DAG}}(q) + \text{size}_{\text{DAG}}\left(\bigcup_x x\sigma\right)\right)\right) \leq \\
\leq P_{\ulcorner\urcorner}\left(n \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2 + \sum_{E \triangleright t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} \sum_x \text{size}_{\text{DAG}}(x\sigma)\right) \leq \\
\leq P_{\ulcorner\urcorner}\left(n \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2 + 2 \times \text{size}_{\text{DAG}}(\mathcal{S}) \times \sum_{E \triangleright t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} \sum_x 1\right) \leq \\
\leq P_{\ulcorner\urcorner}\left(n \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2 + 2 \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2 \times \sum_{E \triangleright t \in \mathcal{S}} \sum_{q \in E \cup \{t\}} 1\right) \leq \\
\leq P_{\ulcorner\urcorner}\left(n \times (\text{size}_{\text{DAG}}(\mathcal{S}))^2 + 2 \times (\text{size}_{\text{DAG}}(\mathcal{S}))^3 \times n\right) \leq \\
\leq P_{\ulcorner\urcorner}\left(3 \times (n \times \text{size}_{\text{DAG}}(\mathcal{S}))^3\right) = O\left((n \times \text{size}_{\text{DAG}}(\mathcal{S}))^{3m''}\right).
\end{aligned}
$$

Thus, $T = O\left((n \times \text{size}_{\text{DAG}}(\mathcal{S}))^{3m''} + (n \times \text{size}_{\text{DAG}}(\mathcal{S}))^{2m'}\right)$ that shows, that a test of a proof returned by the algorithm takes polynomial time what gives us a class of complexity. □

4.2 Ground derivability in DY+ACI is in P

Proposition 12. *Algorithm 2 has a polynomial complexity on* $\text{size}_{\text{DAG}}(E \cup \{t\})$.

Proof. We will give a very coarse estimate.

First remark, that in any step of algorithm, $|S|$ and $|D|$ don't exceed $|\text{QSub}(E \cup \{t\})|$.

Building of sets $\text{QSub}(E) \cup \text{QSub}(t)$ takes linear time on $\text{size}_{\text{DAG}}(E \cup \{t\})$. Building S will take not more than $O(|E| \times |\text{QSub}(E) \cup \text{QSub}(t)|)$, that is, not more than $O\left((\text{size}_{\text{DAG}}(E \cup \{t\}))^2\right)$.

The main loop has at most $|\text{QSub}(E \cup \{t\})| - |E|$ steps. Searching for DY rule with left-hand side in D and right-hand side in S is not greater that $O(|S| \times |D|^2)$ and thus, not greater that $O((\text{size}_{\text{DAG}}(E \cup \{t\}))^3)$. The next `if` can be performed in $O(|S| \times |D| \times (\text{size}_{\text{DAG}}(E \cup \{t\})))$ steps and the last `if` can be also done for cubic time. The check done in `return` statement is linear. And finally, thanks to the Statement 15 of Lemma 3, we can easily justify the claimed complexity. □

5 Satisfiability of general DY constraint system

The previous result on constraint solving for DY+ACI theory can be projected to the classical DY case. We cannot apply it directly, as in the resulting solution we will probably have an ACI symbol. Thus, we need to prove the decidability of DY case. The scheme we follow to solve a constraint system within DY deduction system is shown in Figure 5.

First, we can 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 solution of a given constraint system within DY+ACI.

Third, we will transform the solution obtained from previous step (which is in DY+ACI) in such a way, that the resulting substitution will be a solution of initial constraint system within DY(Theorem 2). The idea of satisfactory transformation δ is simple: we replace any ACI list of terms with nested pairs: $\cdot(\{t_1, \ldots, t_n\})$ we replace with $\text{pair}(t_1, \text{pair}(..., t_n))$. Note, that this transformation will have a linear complexity and the transformed solution will have the DAG-size not more than twice bigger than initial. This gives us a class of complexity, which is NP, for the problem of satisfiability of general constraint system within DY model.

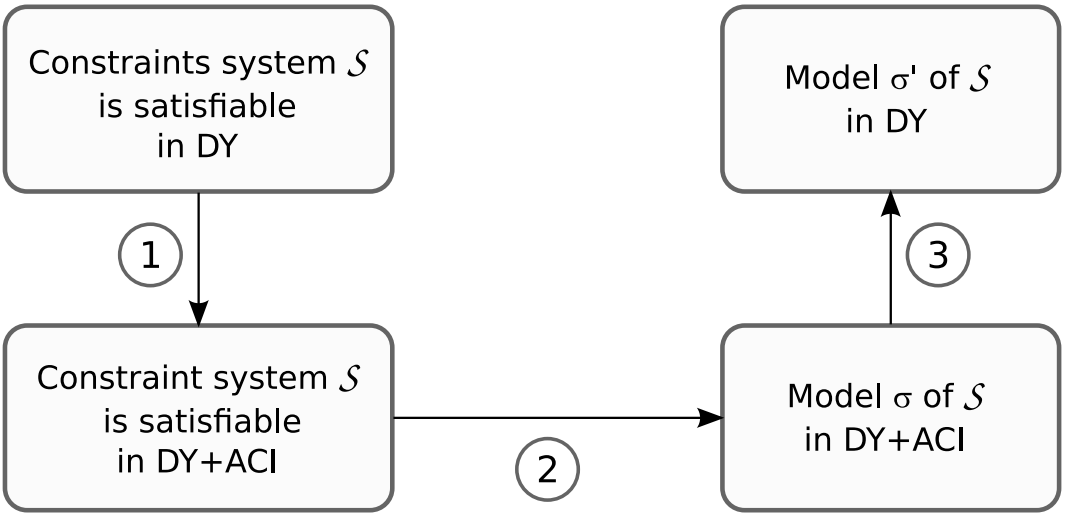

Figure 5: Proof Plan

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

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

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

Let us recall classical Dolev-Yao deduction system (DY) in Table 3.

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

Table 3: DY deduction system rules

Definition 5.3. *A constraint system* $\mathcal{S}$ standard*, if* $\forall s \in \mathrm{Sub}(\mathcal{S})\ \mathrm{root}(s) \neq \cdot$*. 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 17. *Any standard constraint system is normalized.*

Lemma 18. *Let t be a standard term, σ be a normalized substitution. Then $t\sigma$ is normalized.*

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

Proof. It is enough to consider the same solution 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 solution in DY, that σ stays a solution of $\mathcal{S}$ in DY+ACI. □

The goal of the following reasoning is to show that we can build a solution of a constraint system within DY from a solution of this constraint system within DY+ACI.

Lemma 19. *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:

- $t_1, t_2 \to \ulcorner \mathrm{pair}(t_1, t_2) \urcorner$

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

- $t_1, t_2 \to \ulcorner \mathrm{enc}(t_1, t_2) \urcorner$. Proof of this case can be done by analogy of previous one.

- $t_1, t_2 \to \ulcorner \mathrm{aenc}(t_1, t_2) \urcorner$. Proof of this case can be done by analogy of previous one.

- $t_1, \mathrm{priv}(t_2) \to \ulcorner \mathrm{sig}(t_1, \mathrm{priv}(t_2)) \urcorner$.

  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 $\mathrm{priv}(\delta(t_2)) = \delta(\mathrm{priv}(t_2))$.

- $t_1, \ldots, t_m \to \ulcorner \cdot (t_1, \ldots, t_m) \urcorner$.

  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 3.

  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 (proof by induction on 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 $\delta(\text{elems}(t_i))$ for all i, and then rebuild (derive with composition rules) $\delta(\ulcorner \cdot(t_1, \ldots, t_m) \urcorner)$.

- $\text{enc}(t_1, t_2), \ulcorner t_2 \urcorner \to \ulcorner t_1 \urcorner$.

  As $\text{enc}(t_1, t_2)$ is normalized, then $t_1 = \ulcorner t_1 \urcorner$ and $t_2 = \ulcorner t_2 \urcorner$.
  Thus, $\delta(t_1) \in \text{Der}_{DY}(\{\text{enc}(\delta(t_1), \delta(t_2)), \delta(t_2)\})$ and this is what we need, because $\delta(\text{enc}(t_1, t_2)) = \text{enc}(\delta(t_1), \delta(t_2))$.

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

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

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

- $\cdot(t_1, \ldots, t_m) \to \ulcorner t_i \urcorner$.

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

□

Proposition 14. *Given a standard constraint system $\mathcal{S}$ and its normalized solution σ in DY+ACI. Then, for any subterm of the system $t \in \text{Sub}(\mathcal{S})$, we have $\delta(t\sigma) = t\delta(\sigma)$.*

Proof. The proof is done by induction as in Proposition 5.

- Let $\text{size}_{\text{DAG}}(t) = 1$. Then either $t \in \mathcal{A}$ or $t \in \mathcal{X}$. Both are trivial cases.

- Assume that for some $k \geq 1$ if $\text{size}_{\text{DAG}}(t) \leq k$, then $\delta(t\sigma) = t\delta(\sigma)$.

- Show, that for t such that $\text{size}_{\text{DAG}}(t) \geq k+1$, where $t = \text{bin}(p, q)$ or $t = \text{priv}(p)$ and $\text{size}_{\text{DAG}}(p) \leq k$ and $\text{size}_{\text{DAG}}(q) \leq k$, statement $\delta(t\sigma) = t\delta(\sigma)$ is still true. We have:

  - either $t = \text{bin}(p, q)$. As $\delta(\text{bin}(p, q)\sigma) = \delta(\text{bin}(p\sigma, q\sigma)) = \text{bin}(\delta(p\sigma), \delta(q\sigma)) = \text{bin}(p\delta(\sigma), q\delta(\sigma)) = \text{bin}(p, q)\delta(\sigma)$.
  - or $t = \text{priv}(p)$. 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. *Given a standard constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ and its normalized solution σ in DY+ACI. Then $\delta(\sigma)$ is a solution in DY of $\mathcal{S}$.*

Proof. Let $E \rhd t$ be any element of $\mathcal{S}$. As σ is a solution of $\mathcal{S}$, then $\ulcorner t\sigma \urcorner \in \mathrm{Der}\left(\ulcorner E\sigma \urcorner\right)$. As σ is normalized and $\mathcal{S}$ is standard, using Lemma 18 we have $\ulcorner t\sigma \urcorner = t\sigma$ and $\ulcorner E\sigma \urcorner = E\sigma$. Then, $t\sigma \in \mathrm{Der}\left(E\sigma\right)$. 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 Lemma 19 and Lemma 2 (which also works for DY case) we can easily prove that if $k > 0$, $\delta\left(A_j\right) \subseteq \mathrm{Der}_{DY}\left(\delta\left(A_{j-1}\right)\right)$, $j = 1, \ldots, k$. Note, that $\delta\left(A\right)$ is a set of standard terms (and thus, normalized) for any set of terms A. Then, applying transitivity of $\mathrm{Der}_{DY}\left(\cdot\right)$ (Lemma 1 for DY) k times, we have that $\delta\left(A_k\right) \subseteq \mathrm{Der}_{DY}\left(\delta\left(A_0\right)\right)$. In the case where $k = 0$, the statement $\delta\left(A_k\right) \subseteq \mathrm{Der}_{DY}\left(\delta\left(A_0\right)\right)$ is also true.

Using Proposition 14 we have that $\delta\left(A_0\right) = \delta\left(E\sigma\right) = E\delta\left(\sigma\right)$, as $E \subseteq \mathrm{QSub}\left(\mathcal{S}\right)$. The same for t: $\delta\left(t\sigma\right) = t\delta\left(\sigma\right)$, and as $t\sigma \in A_k$, we have $t\delta\left(\sigma\right) \in \delta\left(A_k\right)$.

Thus, we have that $t\delta\left(\sigma\right) \in \delta\left(A_k\right) \subseteq \mathrm{Der}_{DY}\left(\delta\left(A_0\right)\right) = \mathrm{Der}_{DY}\left(E\delta\left(\sigma\right)\right)$, that means $\delta\left(\sigma\right)$ 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 one in Example 5.*

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

Using Algorithm 3, we can get a solution of $\mathcal{S}$ within DY+ACI, let's say, as in Example 6, $\sigma = \{x \mapsto \cdot\left(\{a, b, c\}\right)\}$.

Then, by applying transformation $\delta\left(\cdot\right)$, we will get $\sigma' = \delta\left(\sigma\right) = \{x \mapsto \mathrm{pair}\left(a, \mathrm{pair}\left(b, c\right)\right)\}$. We can see, that σ' is also a solution of $\mathcal{S}$ within DY(as it was proven in Theorem 2).

Corollary 3 (of Theorem 2 and Proposition 13)**.** *A standard constraint system $\mathcal{S}$ is satisfiable within DY iff it is satisfiable within DY+ACI.*

Corollary 4. *Satisfiability of constraint system within DY is in NP.*

6 Conclusions

In this work we have presented a decision algorithm for solving Dolev Yao general constraints as well as ones extended with an ACI symbol, which can be used to represent sets of terms. The complexity of the algorithm was proved to be in NP.

We have given also three applications of the presented result: protocol insecurity with non-communicating intruders, protocol synthesis and discovering XML-based attacks.

Further work is aimed to extend the results to other deduction systems and equality theories.

APPENDIX

A General constraints for subterm theories

Let us call a subterm deduction system a set of rules of two forms:

- composition rules: for all public functional symbols f, $x_1, \ldots, x_k \to f(x_1, \ldots, x_k)$
- 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 constraint system within 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 configuration as terms we shall introduce a set of variables $\mathcal{X}$ and an alphabet $\mathcal{F}$

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

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

The TM configuration with tape $\perp$ *abcde* $\perp$, (where $\perp$ is an endmarker), with symbol d under the head, and state q will be represented by the following term of $q(c(b(a(\perp), d(e(\perp), x)$ 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:

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

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

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

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

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

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:

$$q_1(\bot, \bot, x) \rhd q_n(y, z, w)$$

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

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

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

We should note by contrast (see [3]), 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.

References

[1] David Basin, Sebastian Mödersheim, and Luca Viganò. Algebraic intruder deductions. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, pages 549–564, 2005.

[2] David Basin, Sebastian Mödersheim, and Luca Viganò. Ofmc: A symbolic model checker for security protocols. *International Journal of Information Security*, 4:181–208, June 2005.

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

[4] Sergiu Bursuc. *Contraintes de déductibilité dans une algèbre quotient: réduction de modèles et applications à la sécurité.* Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2009.

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

[6] Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with diffie-hellman exponentiation and products in exponents. In *FSTTCS*, pages 124–135, 2003.

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

[8] Yannick Chevalier, Denis Lugiez, and Michaël Rusinowitch. Towards an automatic analysis of web service security. In *FroCoS 2007, Liverpool, UK, September 10-12*, volume 4720 of *Lecture Notes in Computer Science*, pages 133–147. Springer, 2007.

[9] Yannick Chevalier and Michael Rusinowitch. Symbolic protocol analysis in the union of disjoint intruder theories: Combining decision procedures. *Theoretical Computer Science*, 411(10):1261 – 1282, 2010. ICALP 2005 - Track C: Security and Cryptography Foundations.

[10] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. pages 271–280. IEEE Comp. Soc. Press, 2003.

[11] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.

[12] C.J.F. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, USA, Proc.*, volume 5123/2008 of *Lecture Notes in Computer Science*, pages 414–418. Springer, 2008.

[13] Stéphanie Delaune. *Vérification des protocoles cryptographiques et propriétés algébriques*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2006.

[14] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis for monoidal equational theories. *Information and Computation*, 206(2-4):312–351, February-April 2008.

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

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

[17] Laurent Mazaré. *Computational Soundness of Symbolic Models for Cryptographic Protocols*. PhD thesis, Institut National Polytechnique de Grenoble, October 2006.

[18] Jonathan Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 166–175, New York, NY, USA, 2001. ACM.

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

[20] Vitaly Shmatikov. Decidable analysis of cryptographic protocols with products and modular exponentiation. In *In Proc. 13th European Symposium on Programming (ESOP '04), volume 2986 of LNCS*, pages 355–369. Springer-Verlag, 2004.

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

[22] Mathieu Turuani. The CL-Atse Protocol Analyser. In *Term Rewriting and Applications (RTA)*, pages 277–286, 2006.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399