



**HAL**  
open science

# Knowledge Problems in Security Protocols: Going Beyond Subterm Convergent Theories

Saraid Dwyer Satterfield, Serdar Erbatur, Andrew M. Marshall, Christophe Ringeissen

► **To cite this version:**

Saraid Dwyer Satterfield, Serdar Erbatur, Andrew M. Marshall, Christophe Ringeissen. Knowledge Problems in Security Protocols: Going Beyond Subterm Convergent Theories. 8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023), Jul 2023, Rome, Italy. pp.30:1–30:19, 10.4230/LIPIcs.FSCD.2023.30 . hal-04214220

**HAL Id: hal-04214220**

**<https://inria.hal.science/hal-04214220>**

Submitted on 5 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License


# Knowledge Problems in Security Protocols: Going Beyond Subterm Convergent Theories

Saraid Dwyer Satterfield

University of Mary Washington, Fredericksburg, VA, USA

Serdar Erbatur 

University of Texas at Dallas, TX, USA

Andrew M. Marshall 

University of Mary Washington, Fredericksburg, VA, USA

Christophe Ringeissen 

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

---

## Abstract

We introduce a new form of restricted term rewrite system, the graph-embedded term rewrite system. These systems, and thus the name, are inspired by the graph minor relation and are more flexible extensions of the well-known homeomorphic-embedded property of term rewrite systems. As a motivating application area, we consider the symbolic analysis of security protocols, and more precisely the two knowledge problems defined by the deduction problem and the static equivalence problem. In this field restricted term rewrite systems, such as subterm convergent ones, have proven useful since the knowledge problems are decidable for such systems. However, many of the same decision procedures still work for examples of systems which are “beyond subterm convergent”. However, the applicability of the corresponding decision procedures to these examples must often be proven on an individual basis. This is due to the problem that they don’t fit into an existing syntactic definition for which the procedures are known to work. Here we show that many of these systems belong to a particular subclass of graph-embedded convergent systems, called contracting convergent systems. On the one hand, we show that the knowledge problems are decidable for the subclass of contracting convergent systems. On the other hand, we show that the knowledge problems are undecidable for the class of graph-embedded systems.

**2012 ACM Subject Classification** Theory of computation → Equational logic and rewriting; Theory of computation → Automated reasoning

**Keywords and phrases** Term rewriting, security protocols, verification

**Digital Object Identifier** 10.4230/LIPIcs.FSCD.2023.30

**Funding** *Christophe Ringeissen*: This work has been partly supported by the ANR Research and teaching chair in AI ASAP (ANR-20-CHIA-0024) and the Région Grand Est.

**Acknowledgements** We would like to thank Steve Kremer for his comments on the paper as they were very helpful in improving several technical results.

## 1 Introduction

In this paper we introduce a new form of term rewrite system, called the graph-embedded term rewrite systems, and motivate the study and use of such rewrite systems by demonstrating their usefulness in the application of security protocols.

The research area of cryptographic protocol analysis contains a number of innovative algorithms and procedures for checking various security properties of protocols, see for example [1, 12, 15, 17]. These procedures consider protocols modeled in a symbolic way, typically via a rewrite system or equational theory. Often the procedure is proven sound



© Saraid Dwyer Satterfield, Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen; licensed under Creative Commons License CC-BY 4.0

8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023).

Editors: Marco Gaboardi and Femke van Raamsdonk; Article No. 30; pp. 30:1–30:19



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and complete for specific classes of theories. One of the most common classes are those theories that can be represented by subterm convergent term rewrite systems. That is, term rewrite systems where the right-hand side of the rules are ground or strict subterms of the left-hand side. For example, see the procedures developed in [1, 15]. Interestingly, many of these same procedures also work for theories that are “beyond subterm convergent”. That is, they are not strictly subterm convergent. However, since these examples don’t fit into a known class of theories for which soundness and completeness proofs already exist, they must be proven on an individual bases. For example, the procedures of [1, 12, 15, 17] are shown to work on the theory of blind signatures, see Example 2 below. However, the theory is not subterm convergent, notice in the final rule,  $unblind(sign(blind(x, y), z), y) \rightarrow sign(x, z)$ , that  $sign(x, z)$  is not a strict subterm of  $unblind(sign(blind(x, y), z), y)$ . Thus, in each case a unique proof is needed to show applicability of the procedure on the theory of blind signatures. Several additional examples of beyond subterm theories are given throughout the paper. This begs the question of whether there is a syntactic definition of a class of term rewrite systems such that the definition encapsulates these beyond subterm examples yet still maintains some of the useful properties needed to ensure applicability of the above procedures.

In the paper we answer the question in the positive by introducing first graph-embedded term rewrite systems and then a particular subclass called contracting rewrite systems. These systems are inspired by the notions of graph embeddings and graph minors. Here we are able to translate the notion to term rewrite systems. This translation is done in a very similar fashion to what has been done with homeomorphic embeddings. We are able to provide a rewrite schema which induces graph-embedded systems in a similar way in which homeomorphic-embedded systems are induced by a rewrite system (see [7] for more details). To the best of our knowledge these systems have not been explored before. We then explore some of the properties of these new systems. Interestingly, the graph-embedded systems encompass most of the beyond subterm examples from many of the protocol analysis procedures [1, 12, 15, 17]. As an initial step, in this paper we concentrate on the knowledge problems considered in [1] using the notion of locally stable theories. Local stability is a desirable property which ensures the decidability of the critical symbolic security question of deducibility. In the class of graph-embedded convergent systems, we are now able to identify a particular subclass called the contracting convergent systems, which are beyond subterm convergent, encompass most of the beyond subterm examples of [1, 12, 15, 17], and are locally stable. As a consequence, the knowledge problems of deduction and static equivalence are decidable for the subclass of contracting convergent systems. Furthermore, we show that the knowledge problems are undecidable for the class of graph-embedded convergent systems in general.

Finally, this paper represents the initial exploration of graph-embedded term rewrite systems and their application to protocol analysis. We hope that the formulation proves useful in areas beyond security protocols as homeomorphic embeddings have proven useful in many areas. We conclude the paper with a discussion of several open questions related to graph-embedded systems.

**Paper Outline.** The remainder of the paper is organized as follows. Section 2 contains the preliminaries, introducing the necessary background material on term-rewrite systems, graph theory and security protocol analysis. Section 3 introduces the graph-embedded term rewrite systems and explores some of their basic properties. Section 4 introduces the motivating application area of this paper for graph-embedded systems, security protocol analysis. In that section, we show that the knowledge problems are undecidable for the

class of graph-embedded convergent systems but decidable for the subclass of contracting convergent systems. Section 5 considers the relation to another common and useful property, the Finite Variant Property (FVP). Finally, Section 6 contains the concluding remarks, future work, and some open problems. Our decidability result relies on lemmas that are proven in Appendix A.

## 2 Preliminaries

We use the standard notation of equational unification [8] and term rewriting systems [7]. Given a first-order signature  $\Sigma$  and a (countable) set of variables  $V$ , the  $\Sigma$ -terms over variables  $V$  are built in the usual way by taking into account the arity of each function symbol in  $\Sigma$ . Each  $\Sigma$ -term is well-formed: if it is rooted by a  $n$ -ary function symbol in  $\Sigma$ , then it has necessarily  $n$  direct subterms. The set of  $\Sigma$ -terms over variables  $V$  is denoted by  $T(\Sigma, V)$ . Given a (countable) set of constants  $C$  disjoint from  $V$  and  $\Sigma$ , the set of  $\Sigma$ -terms over  $V \cup C$  is denoted in the same way by  $T(\Sigma, V \cup C)$ . In the following, a  $\Sigma$ -term is assumed to be a term in  $T(\Sigma, V \cup C)$ . The set of variables (resp., constants) from  $V$  (resp.,  $C$ ) occurring in a term  $t \in T(\Sigma, V \cup C)$  is denoted by  $Var(t)$  (resp.,  $Cst(t)$ ). A term  $t$  is *ground* if  $Var(t) = \emptyset$ . A  $\Sigma \cup C$ -rooted term is a term whose root symbol is in  $\Sigma \cup C$ . For any position  $p$  in a term  $t$  (including the root position  $\epsilon$ ),  $t(p)$  is the symbol at position  $p$ ,  $t|_p$  is the subterm of  $t$  at position  $p$ , and  $t[u]_p$  is the term  $t$  in which  $t|_p$  is replaced by  $u$ . A substitution is an endomorphism of  $T(\Sigma, V \cup C)$  with only finitely many variables not mapped to themselves. A substitution is denoted by  $\sigma = \{x_1 \mapsto t_1, \dots, x_m \mapsto t_m\}$ , where the domain of  $\sigma$  is  $Dom(\sigma) = \{x_1, \dots, x_m\}$  and the range of  $\sigma$  is  $Ran(\sigma) = \{t_1, \dots, t_m\}$ . Application of a substitution  $\sigma$  to  $t$  is written  $t\sigma$ .

The size of a term  $t$ , denoted by  $|t|$ , is defined inductively as follows:  $|f(t_1, \dots, t_n)| = 1 + \sum_{i=1}^n |t_i|$  if  $f$  is a  $n$ -ary function symbol with  $n \geq 1$ ,  $|c| = 1$  if  $c$  is a constant, and  $|x| = 1$  if  $x$  is a variable. The depth of a term  $t$ , denoted by  $depth(t)$ , is defined inductively as follows:  $depth(f(t_1, \dots, t_n)) = 1 + \max_{i=1, \dots, n} depth(t_i)$  if  $f$  a  $n$ -ary function symbol with  $n \geq 1$ ,  $depth(c) = 0$  if  $c$  is a constant, and  $depth(x) = 0$  if  $x$  is a variable.

A *context* is a term with holes. More formally, a context is a term where each variable occurs at most once. Thus, the size of a context follows from the size of a term, where any hole occurrence counts for 1.

### Equational Theories

Given a set  $E$  of  $\Sigma$ -axioms (i.e., pairs of terms in  $T(\Sigma, V)$ , denoted by  $l = r$ ), the *equational theory*  $=_E$  is the congruence closure of  $E$  under the law of substitutivity (by a slight abuse of terminology,  $E$  is often called an equational theory). Equivalently,  $=_E$  can be defined as the reflexive transitive closure  $\leftrightarrow_E^*$  of an equational step  $\leftrightarrow_E$  defined as follows:  $s \leftrightarrow_E t$  if there exist a position  $p$  of  $s$ ,  $l = r$  (or  $r = l$ ) in  $E$ , and substitution  $\sigma$  such that  $s|_p = l\sigma$  and  $t = s[r\sigma]_p$ .

### Rewrite Relations

A *term rewrite system* (TRS) is a pair  $(\Sigma, R)$ , where  $\Sigma$  is a signature and  $R$  is a finite set of rewrite rules of the form  $l \rightarrow r$  such that  $l, r$  are  $\Sigma$ -terms,  $l$  is not a variable and  $Var(r) \subseteq Var(l)$ . A term  $s$  *rewrites* to a term  $t$  w.r.t  $R$ , denoted by  $s \rightarrow_R t$  (or simply  $s \rightarrow t$ ), if there exist a position  $p$  of  $s$ ,  $l \rightarrow r \in R$ , and substitution  $\sigma$  such that  $s|_p = l\sigma$  and  $t = s[r\sigma]_p$ . If the rewrite step occurs at the root position of the term  $s$  we denote this

as  $s \rightarrow_R^\epsilon t$ . When  $\sigma$  is a variable renaming, we say that  $s$  rewrites to  $t$  applying a *variable instance* of  $l \rightarrow r$ . A TRS  $R$  is *terminating* if there are no infinite reduction sequences with respect to  $\rightarrow_R$ . A TRS  $R$  is *confluent* if, whenever  $t \rightarrow_R^* s_1$  and  $t \rightarrow_R^* s_2$ , there exists a term  $w$  such that  $s_1 \rightarrow_R^* w$  and  $s_2 \rightarrow_R^* w$ . A confluent and terminating TRS is called *convergent*. In a convergent TRS  $R$ , we have the existence and the uniqueness of  $R$ -normal forms, denoted by  $t \downarrow_R$  for any term  $t$ . When  $R$  is clear from the context, the normal form of  $t$  may be written  $t \downarrow$ . Given a substitution  $\sigma$ ,  $\sigma \downarrow = \{x \mapsto (x\sigma) \downarrow\}_{x \in \text{Dom}(\sigma)}$  is the substitution corresponding to the normal form of  $\sigma$ .

A convergent *term rewrite system* (TRS)  $R$  is said to be *subterm convergent* if for any  $l \rightarrow r \in R$ ,  $r$  is either a strict subterm of  $l$  or a ground term. An equational theory,  $E$ , is *subterm convergent* if it is presented by a subterm convergent TRS. That is, there exists a subterm convergent TRS,  $R$ , such that  $=_E$  and  $=_R$  coincide.

► **Definition 1** (Homeomorphic Embedding). *The homeomorphic embedding,  $\succeq_{emb}$  is a binary relation on terms such that:  $s \succeq_{emb} t$  if one of the following conditions hold:*

1.  $s = x = t$  for some variable  $x$ ,
2.  $s = f(s_1, \dots, s_n)$  and  $t = f(t_1, \dots, t_n)$  and  $s_1 \succeq_{emb} t_1, \dots, s_n \succeq_{emb} t_n$ ,
3.  $s = f(s_1, \dots, s_n)$  and  $s_i \succeq_{emb} t$  for some  $i$ ,  $1 \leq i \leq n$ .

A TRS  $R$  is said to be a *homeomorphic-embedded TRS* if for any  $l \rightarrow r \in R$ ,  $l \succeq_{emb} r$ .

More interestingly we can also define  $\succeq_{emb}$  as the reduction relation  $\rightarrow_{R_{emb}}^*$  induced by the rewrite system  $R_{emb} = \{f(x_1, \dots, x_n) \rightarrow x_i \mid f \text{ is } n\text{-ary}, n \geq 1, 1 \leq i \leq n\}$ .

► **Example 2** (Blind Signatures). The theory of blind signatures [15] is a homeomorphic-embedded convergent TRS:

$$\begin{aligned} \text{checksign}(\text{sign}(x, y), \text{pk}(y)) &\rightarrow x \\ \text{unblind}(\text{blind}(x, y), y) &\rightarrow x \\ \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) &\rightarrow \text{sign}(x, z) \end{aligned}$$

## Notions of Knowledge

The applied pi calculus and frames are used to model attacker knowledge [2]. In this model, the set of messages or terms which the attacker knows, and which could have been obtained from observing one or more protocol sessions, are the set of terms in  $\text{Ran}(\sigma)$  of the frame  $\phi = \nu \tilde{n}. \sigma$ , where  $\sigma$  is a substitution ranging over ground terms. We also need to model cryptographic concepts such as nonces, keys, and publicly known values. We do this by using names, which are essentially free constants. Here also, we need to track the names which the attacker knows, such as public values, and the names which the attacker does not know a priori, such as freshly generated nonces.  $\tilde{n}$  consists of a finite set of restricted names, these names represent freshly generated names which remain secret from the attacker. The set of names occurring in a term  $t$  is denoted by  $\text{fn}(t)$ . For any frame  $\phi = \nu \tilde{n}. \sigma$ , let  $\text{fn}(\phi)$  be the set of names  $\text{fn}(\sigma) \setminus \tilde{n}$  where  $\text{fn}(\sigma) = \bigcup_{t \in \text{Ran}(\sigma)} \text{fn}(t)$ ; and for any term  $t$ , let  $t\phi$  denote by a slight abuse of notation the term  $t\sigma$ . We say that a term  $t$  *satisfies the name restriction* (of  $\phi$ ) if  $\text{fn}(t) \cap \tilde{n} = \emptyset$ .

► **Definition 3** (Deduction). *Let  $\phi = \nu \tilde{n}. \sigma$  be a frame, and  $t$  a ground term. We say that  $t$  is deduced from  $\phi$  modulo  $E$ , denoted by  $\phi \vdash_E t$ , if there exists a term  $\zeta$  such that  $\zeta\sigma =_E t$  and  $\text{fn}(\zeta) \cap \tilde{n} = \emptyset$ . The term  $\zeta$  is called a *recipe of  $t$  in  $\phi$  modulo  $E$* .*

Another form of knowledge is the ability to tell if two frames are *statically equivalent* modulo  $E$ , sometimes also called *indistinguishability*.

► **Definition 4** (Static Equivalence). *Two terms  $s$  and  $t$  are equal in a frame  $\phi = \nu\tilde{n}.\sigma$  modulo an equational theory  $E$ , denoted  $(s =_E t)\phi$ , if  $s\sigma =_E t\sigma$ , and  $\tilde{n} \cap (\text{fn}(s) \cup \text{fn}(t)) = \emptyset$ . The set of all equalities  $s = t$  such that  $(s =_E t)\phi$  is denoted by  $\text{Eq}(\phi)$ . Given a set of equalities  $\text{Eq}$ , the fact that  $(s =_E t)\phi$  for any  $s = t \in \text{Eq}$  is denoted by  $\phi \models \text{Eq}$ . Two frames  $\phi = \nu\tilde{n}.\sigma$  and  $\psi = \nu\tilde{n}.\tau$  are statically equivalent modulo  $E$ , denoted as  $\phi \approx_E \psi$ , if  $\text{Dom}(\sigma) = \text{Dom}(\tau)$ ,  $\phi \models \text{Eq}(\psi)$  and  $\psi \models \text{Eq}(\phi)$ .*

Both deduction and static equivalence are known to be decidable in subterm convergent rewrite systems [1]. In this paper, we lift these results to rewrite systems that are beyond the class of subterm convergent rewrite systems.

► **Example 5.** Let  $E$  be the equational theory presented by the subterm convergent TRS  $\{\text{dec}(\text{enc}(x, y), y) \rightarrow x\}$ . Applying the decision procedure developed in [1], one can check that  $\phi = \nu\{n\}.\{v \mapsto \text{enc}(a, n)\}$  and  $\psi = \nu\{n\}.\{v \mapsto \text{enc}(b, n)\}$  are statically equivalent modulo  $E$ . Consider now  $\phi' = \nu\{n\}.\{v \mapsto \text{enc}(a, n), w \mapsto n\}$  and  $\psi' = \nu\{n\}.\{v \mapsto \text{enc}(b, n), w \mapsto n\}$ . Since  $\text{dec}(v, w) = a \in \text{Eq}(\phi')$  and  $\text{dec}(v, w) = a \notin \text{Eq}(\psi')$ ,  $\phi'$  and  $\psi'$  are not statically equivalent modulo  $E$ .

## Term Graphs

Each term  $t$  can be viewed in a graphical representation, called a *term graph*. Each node in the graph is labeled either by a function symbol or a variable. Each function symbol node also has an associated successor number, corresponding to the arity of the function. Edges connect the nodes of the term graph based on the subterm relation. The notion of term graph is illustrated in Examples 19 and 20.

► **Definition 6** (Term Graph Measures). *We introduce some convenient notation:*

- *Let  $\text{VP}(t)$  denote the list of leaf nodes in the term graph of a term  $t$  labeled by a variable. Notice that two distinct nodes could be labeled by the same variable.*
- *Let  $\text{FP}(t)$  denote the list of nodes in the term graph of  $t$  labeled by a function symbol. Notice that two distinct nodes could be labeled by the same function symbol.*
- *Let  $\text{FS}(t)$  denote the set of function symbols in the term  $t$ .*

## Some Graph Theory

We will also need a few notions from graph theory, we introduce those in this section. We will typically use  $G$  to denote a graph,  $V$  the set of vertex and  $E$  the set of edges of the graph.

► **Definition 7** (Graph Isomorphism). *Let  $G = (V, E)$  and  $G' = (V', E')$  be two graphs. We say that  $G$  and  $G'$  are isomorphic, denoted  $G \simeq G'$ , if there exists a bijection  $\phi : V \rightarrow V'$  with  $xy \in E$  iff  $\phi(x)\phi(y) \in E'$ ,  $\forall x, y \in V$ .*

► **Definition 8** (Edge Contraction). *Let  $G = (V, E)$  and  $e = xy$ .  $G/e$  is the graph  $G' = (V', E')$  such that  $V' = (V \setminus \{x, y\}) \cup \{v_e\}$ , where  $v_e$  is a new vertex, and  $E' = \{vw \in E \mid \{v, w\} \cap \{x, y\} = \emptyset\} \cup \{v_e w \mid xw \in E \setminus \{e\} \text{ or } yw \in E \setminus \{e\}\}$ .*

*We say that  $G'$  is obtained from  $G$  by contracting the edge  $e$ .*

We use the following definition of graph minor which essentially says that a graph minor of a graph  $G$  can be obtained by a series of graph contractions (see [16] for more details).

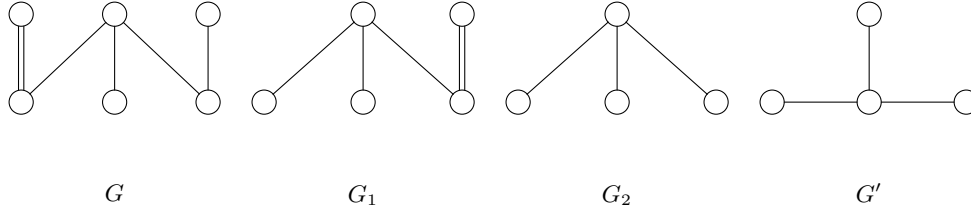
► **Definition 9** (Graph Minor). *The graph  $G'$  is a graph minor of the graph  $G$ , if there exist graphs  $G_0, G_1, \dots, G_n$  and edges  $e_i \in G_i$  such that  $G = G_0, G_n \simeq G'$ , and  $G_{i+1} = G_i/e_i$  for  $i = 0, \dots, n-1$ . We use the notation  $G \succcurlyeq G'$  if  $G'$  is a graph minor of  $G$ .*

*Further, we extend the pure graph definition above to terms using the same notion as follows. For terms  $t$  and  $t'$ ,  $t \succcurlyeq t'$  if for the corresponding term graphs of  $t$  and  $t'$ , denoted as  $G$  and  $G'$  respectively, we have  $G \succcurlyeq G'$ .*

► **Remark 10.** Note, in the classical definition of graph minor (see [16]), if  $G$  is a subgraph of a larger graph  $G_{large}$ , then also  $G_{large} \succcurlyeq G'$ . However, this component of the definition is not necessary for the results of this paper and by leaving it out we are able to simplify the later definitions and presentation.

The above type of embedding, denoted by  $\succcurlyeq$ , provides more flexibility than the traditional subterm relation while still preserving some features we need.

► **Example 11.** Notice that  $G'$  is obtained from  $G$  by first applying a sequence of edge contractions, contracting the edge depicted by  $\parallel$  at each step, resulting in  $G_2$ , and finally  $G_2 \simeq G'$ . Therefore,  $G \succcurlyeq G'$ .



We can now extend the above graph-theoretic notions to the term rewrite setting.

### 3 Graph-Embedded Systems

The key to translating from the graph theory setting to the term setting is to use the same methods, *contractions*, but require that the *final* term graph constructed in this fashion represent a *well-formed* term. That is, we need to enforce the notion of a well formed term.

To begin we need to model the graph isomorphism. A restricted form of isomorphism can be translated into the term rewriting setting by considering permutations.

► **Definition 12** (Leaf and Subterm Permutations). *We define two types of permutations,  $\approx_s$  and  $\approx_l$ :*

1. *For terms  $t$  and  $t'$ , we say  $t$  is subterm permutatively equal to  $t'$ , denoted  $t \approx_s t'$ , if one of the following is true:*
  - a.  $t = t'$ , where  $t$  and  $t'$  are constants or variables, or
  - b.  $t = f(u_1, \dots, u_n)$  and  $t' = f(u_{\sigma(1)}, \dots, u_{\sigma(n)})$  where  $f$  is a  $n$ -ary function symbol,  $n \geq 1$ , and  $\sigma$  is a permutation of the indexes  $(1, \dots, n)$ .
2. *For terms  $t$  and  $t'$ , we say  $t$  is leaf permutatively equal to  $t'$ , denoted  $t \approx_l t'$ , if  $t' = t\sigma$  and  $\sigma$  is the unique endomorphism of  $T(\Sigma \cup V \cup C)$  such that its restriction to  $\text{Var}(t) \cup \text{Cst}(t)$  is a permutation on  $\text{Var}(t) \cup \text{Cst}(t)$  and its restriction to  $(V \cup C) \setminus (\text{Var}(t) \cup \text{Cst}(t))$  is the identity.*

The first type of permutation,  $\approx_s$ , allows for permutation inside the term but preserves the layer like structure of the function symbols in the term graph. The second type of permutation in the classical leaf permutability and is restricted to the leaf nodes, i.e., just the variables and constants of the term graph. We will use a combination of the above two permutations in the definition employed for graph-embedded TRS.



► **Definition 13** (Permutatively Equal). *For terms  $t$  and  $t'$ , we say  $t$  is permutatively equal to  $t'$ , denoted  $t \approx t'$ , if  $t \approx_s t'' \approx_i t'$ , for some term  $t''$ .*

► **Remark 14.** It is useful here to remark on the motivation of the above definition,  $\approx$ . The goal is to model the graph isomorphism property. At the same time one needs to be careful not to be too broad and remove layer preserving properties of Definition 15 and thus later protocol properties such as local stability (see Definition 46). In addition, one cannot be too restricted and disallow working protocol representations such as Example 22 which requires more than just leaf permutability. However, it may be possible to improve upon the above definition and allow for additional systems while still maintaining the decidability of the knowledge problems shown here, see the discussion in Section 6.

The next step is to develop a set of rewrite *schema* which preserve a type of graph minor relation on the term graphs. This set of schema then induces a graph-embedded term rewrite system. Notice that this is very similar to what is often done when considering the homeomorphic embeddings, see Definition 1.

► **Definition 15** (Graph Embedding). *Consider the following reduction relation,  $\rightarrow_{R_{gemb}}^*$ , where  $R_{gemb}$  is the set of rewrite rules given by the instantiation of the following rule schema:*

$$\left\{ \begin{array}{l} \text{for any } f \in \Sigma \\ (1) \quad f(x_1, \dots, x_n) \rightarrow x_i \\ (2) \quad f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) \rightarrow f(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \\ \text{and for any } f, g \in \Sigma \\ (3) \quad f(x_1, \dots, x_{i-1}, g(\bar{z}), x_{i+1}, \dots, x_m) \rightarrow g(x_1, \dots, x_{i-1}, \bar{z}, x_{i+1}, \dots, x_m) \\ (4) \quad f(x_1, \dots, x_{i-1}, g(\bar{z}), x_{i+1}, \dots, x_m) \rightarrow f(x_1, \dots, x_{i-1}, \bar{z}, x_{i+1}, \dots, x_m) \end{array} \right\}$$

*We say a term  $t'$  is graph-embedded in a term  $t$ , denoted  $t \succ_{gemb} t'$ , if  $t'$  is a well formed term and there exists a term  $s$  such that  $t \rightarrow_{R_{gemb}}^* s \approx t'$ .*

*A TRS  $R$  is graph-embedded if for any  $l \rightarrow r \in R$ ,  $l \succ_{gemb} r$  or  $r$  is a constant.*

► **Remark 16.** Notice that the rules in  $R_{gemb}$  ignore function arity, thus intermediate terms between  $t$  and  $t'$  may not be well formed. It is only the final term for which function arity and the relation between variables and functions must obey the standard term definition requirements. Note also that, like homeomorphic embedding, the particular schema rules themselves allow for the rewriting of terms down to a single variable. However, the schema are being used to establish the graph-embedded property of a TRS with non-trivial normal forms.

► **Remark 17.** The rules of Definition 15 provide a convenient schema for defining graph-embedded systems. However, they are also very useful in proving properties about graph-embedded systems. Notice that any rewrite step in a graph-embedded system corresponds to one or more steps of the above rules, thus proofs about graph-embedded TRSs can often be reduced to arguments on the properties of the rules of Definition 15.

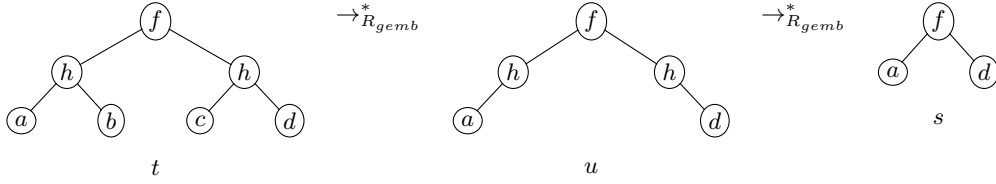
Definition 15 provides a rewrite relation interpretation of graph-embedded systems which is contained in the  $\succ$  relation given in Definition 9.

► **Lemma 18.** *For any terms  $t$  and  $t'$ ,  $t \succ_{gemb} t'$  implies  $t \succ t'$ , i.e.,  $\succ_{gemb} \subseteq \succ$ .*

► **Example 19.** Consider the two terms  $t = f(h(a, b), h(c, d))$  and  $t' = f(d, a)$ . Then,  $t \succ_{gemb} t'$ , since  $t \rightarrow_{R_{gemb}}^* s \approx t'$  where the derivation  $t \rightarrow_{R_{gemb}}^* s$  is as follows:



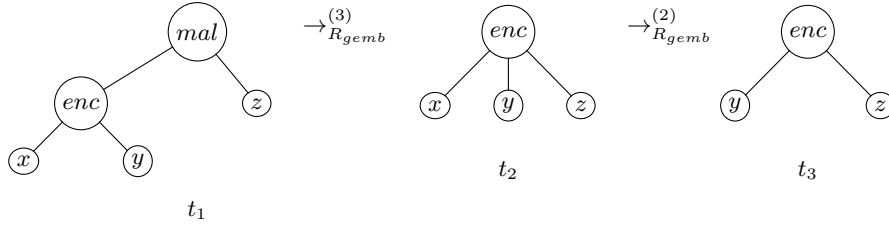
30:8 Going Beyond Subterm Convergent Theories



► **Example 20** (Malleable Encryption). Consider the theory of Malleable Encryption,  $R_{mal}$ :

$$\begin{aligned} dec(enc(x, y), y) &\rightarrow x \\ mal(enc(x, y), z) &\rightarrow enc(z, y) \end{aligned}$$

For the second rule, let  $t_1 = mal(enc(x, y), z)$  and the following derivation  $t_1 \rightarrow^*_{R_{gemb}} t_3$ :



Since  $t_3 \approx enc(z, y)$ , we have  $mal(enc(x, y), z) \succ_{gemb} enc(z, y)$ . The first rule of  $R_{mal}$  being subterm,  $dec(enc(x, y), y) \succ_{gemb} x$ . Thus,  $R_{mal}$  is a graph-embedded TRS.

► **Example 21.** The theory of blind signatures from Example 2 is also a graph-embedded TRS. All but the final rule are subterm. For the final rule,

$$unblind(sign(blind(x, y), z), y) \rightarrow_{R_{gemb}} sign(blind(x, y), z)$$

via rule (1). Then,

$$sign(blind(x, y), z) \rightarrow_{R_{gemb}} sign(x, y, z)$$

via rule (4). Notice again that this intermediate term is not well formed. Finally

$$sign(x, y, z) \rightarrow_{R_{gemb}} sign(x, z) \approx sign(x, z)$$

via rule (2).

► **Example 22** (Addition). Consider the theory of Addition,  $R_{add}$ , from [1]:

$$\begin{aligned} plus(x, s(y)) &\rightarrow plus(s(x), y) \\ plus(x, 0) &\rightarrow x \\ pred(s(x)) &\rightarrow x \end{aligned}$$

$R_{add}$  is a graph-embedded TRS. Notice that  $plus(x, s(y)) \approx plus(s(x), y)$ .

► **Example 23** (Prefix with Pairing). The theory of prefix with pairing [14, 17] is a graph-embedded TRS:

$$\begin{aligned} dec(enc(x, y), y) &\rightarrow x \\ prefix(enc(\langle x, y \rangle, z)) &\rightarrow enc(x, z) \\ fst(\langle x, y \rangle) &\rightarrow x \\ snd(\langle x, y \rangle) &\rightarrow y \end{aligned}$$

► **Example 24** (Trap-door Commitment). The theory of trap-door commitment [15] is a graph-embedded TRS:

$$\begin{aligned} \text{open}(\text{td}(x, y, z), y) &\rightarrow x \\ \text{open}(\text{td}(x_1, y, z), f(x_1, y, z, x_2)) &\rightarrow x_2 \\ \text{td}(x_2, f(x_1, y, z, x_2), z) &\rightarrow \text{td}(x_1, y, z) \\ f(x_2, f(x_1, y, z, x_2), z, x_3) &\rightarrow f(x_1, y, z, x_3) \end{aligned}$$

► **Example 25** (Strong Secrecy). Subterm convergent theories where the right hand side can be a subterm or a constant are graph-embedded such as the following system for considering a form of strong secrecy [10, 12]:

$$\begin{aligned} \text{fst}(\langle x, y \rangle) &\rightarrow x \\ \text{snd}(\langle x, y \rangle) &\rightarrow y \\ \text{adec}(\text{aenc}(x, \text{pk}(y)), y) &\rightarrow x \\ \text{dec}(\text{enc}(x, y), y) &\rightarrow x \\ \text{check}(\text{sign}(x, y), \text{pk}(y)) &\rightarrow \text{ok} \\ \text{msg}(\text{sign}(x, y)) &\rightarrow x \end{aligned}$$

### 3.1 Some Properties of Graph-Embedded Systems

As an initial step we explore some of the basic properties of the graph-embedded TRSs. Similar to the class of subterm TRSs, the graph-embedded TRSs have several nice properties such as termination.

We can first note that the  $\succ_{gemb}$  relation is a partial order on the class of terms. This follows from Lemma 18 and the fact that the graph-embedded relation is a partial ordering on the class of finite graphs (See Proposition 1.7.3 from [16]).

In addition, rewriting at the root position preserves the graph-embedded property. This is due to the fact that for any graph-embedded TRS  $R$  and for any  $l \rightarrow r \in R$ ,  $l \succ_{gemb} r$ . Thus,  $l\sigma = t_1 \succ_{gemb} t_2 = r\sigma$ . More formally, if  $t_1$  and  $t_2$  are terms,  $R$  a graph-embedded TRS, and  $t_1 \rightarrow_R^\epsilon t_2$ , then,  $t_1 \succ_{gemb} t_2$ .

Graph-embedded systems also have the nice property of being size reducing when rewrite steps are applied and thus terminating.

► **Lemma 26.** *Let  $R$  be a graph-embedded TRS such that for all  $l \rightarrow r \in R$ ,  $l \rightarrow_{R_{gemb}}^+ \cdot \approx r$ . Assume  $t \rightarrow_R t'$ . Then,*

- $|Var(t')| \leq |Var(t)|$ ,
- $|VP(t')| \leq |VP(t)|$ ,
- $FS(t') \subseteq FS(t)$ ,
- $|FP(t')| < |FP(t)|$ .

**Proof.** No rule from Definition 15 introduces additional function symbols or variables, satisfying the first and third condition. All rules from Definition 15 remove function symbols except the second rule and  $\approx$ . Notice that if rule 2 is applied then one of the other rules must also be applied to ensure the final term is well formed. Finally, since we require that at least one rewrite step is applied, the size of the term will be reduced even if  $\approx$  doesn't reduce the size of the term. Thus the remaining conditions are satisfied. ◀

► **Remark 27.** Notice that if only  $\approx$  steps are applied in a graph-embedded system then termination is not guaranteed. However, if at least one rewrite rule from  $R_{gemb}$  is applied then by Lemma 26, the system will be terminating.

### Comparing Definitions

We can compare the two embedded definitions. Consider Malleable Encryption,  $R_{mal}$ , from Example 20.  $R_{mal}$  is a graph-embedded TRS, as is shown in Example 20. However,  $R_{mal}$  is not a homeomorphic-embedded TRS. This can be seen in the rule  $mal(enc(x, y), z) \rightarrow enc(z, y)$ . There is no way to obtain the term  $enc(z, y)$  from the term  $mal(enc(x, y), z)$  by application of only the projection rule,  $f(x_1, \dots, x_n) \rightarrow x_i$ . Thus, it's easy to see that there exist graph-embedded TRSs which are not homeomorphic-embedded TRSs. Furthermore, we see that homeomorphic-embedded TRSs are a subset of graph-embedded TRSs.

► **Example 28.** Consider the theory of trap-door commitment from Example 24. Notice that this theory is not a homeomorphic-embedded TRS. For the final rule,

$$f(x_2, f(x_1, y, z, x_2), z, x_3) \rightarrow f(x_1, y, z, x_3),$$

we cannot obtain the right-hand side from the left by the simple projection type relation of Definition 1.

## 4 Knowledge Problems in Graph-Embedded Systems

In this section we look at how graph-embedded TRSs can be used to both extend results in security protocols and also give a formal syntactic definition to classes of protocol presentations for which the decidability of the two knowledge problems are already known. In this section we focus on theories with the local stability property as introduced in [1] (and extended in [6]). For this purpose, we need to consider a restricted form of graph-embedded system called contracting system introduced in Definition 33. One can show that without such a restriction, the knowledge problems for graph-embedded TRSs are undecidable in general.

### 4.1 Undecidable Knowledge Problems

It is shown in [1] that the knowledge problems are undecidable in general. For graph-embedded systems we can reuse, with modification, a proof that was developed in [3] for the unification problem in  $\Delta$ -strong convergent theories. A very similar proof can be found in the research report [4] of the paper [5]. The proof in [3] is via a reduction from the Modified Post Correspondence Problem (MPCP). There, MPCP is needed to ensure solutions, such as  $\sigma = \{x \mapsto c, y \mapsto c\}$ , which don't actually solve an instance of the problem are not possible. We use a similar approach but reduce from the standard Post Correspondence Problem (PCP), using a similar rewrite system to that developed in [3]. We use the standard PCP instead of the MPCP since the encoding works better for the deduction problem where you are finding recipe terms not substitutions. This modification also requires adapting the reduction to the use of frames.

Let  $\Gamma = \{a, b\}$  be the alphabet of the PCP problem. Then, an instance of the problem is a finite set of string pairs,  $S = \{(\alpha_i, \beta_i) \mid i \in [1, n]\} \subseteq \Gamma^+ \times \Gamma^+$ . A solution is a sequence of indexes  $i_1, \dots, i_k \in [1, n]$  such that  $\alpha_{i_1}\alpha_{i_2}\dots\alpha_{i_k} = \beta_{i_1}\beta_{i_2}\dots\beta_{i_k}$ .

► **Lemma 29.** *The deduction problem is undecidable for the class of homeomorphic-embedded convergent TRSs.*

**Proof.** Let  $PCP = \{(\alpha_i, \beta_i) \mid i \in [1, n]\}$  over the alphabet  $\Gamma = \{a, b\}$ . Consider unary function symbols  $a_1, b_1, a_2, b_2$ , and  $g_i$  for each  $i \in [1, n]$ . Let  $f$  be a ternary function symbol and let  $c$  be a constant. Each string from  $PCP$  can be viewed as a sequence of applications of the unary function symbols. Let  $\gamma \in \Gamma$  and for each pair  $(\alpha_i, \beta_i)$ ,  $\tilde{\alpha}_i(x) = \widetilde{\gamma\alpha'_i(x)} = \gamma_1(\tilde{\alpha}'_i(x))$ , and  $\tilde{\beta}_i(x) = \widetilde{\gamma\beta'_i(x)} = \gamma_2(\tilde{\beta}'_i(x))$ . Construct a TRS  $R$  as follows: Let  $R = \bigcup_{i=1}^n \{f(\tilde{\alpha}_i(x), g_i(y), \tilde{\beta}_i(z)) \rightarrow f(x, y, z)\}$ . The  $g_i$  ensure there are no critical pairs between rules and thus we have a convergent TRS. Consider the frame  $\phi = \nu\tilde{n}.\sigma$  with  $\tilde{n} = \{c\}$  and  $\sigma = \bigcup_{i=1}^n \{x_i \mapsto \tilde{\alpha}_i(c), y_i \mapsto \tilde{\beta}_i(c), z_i \mapsto g_i(c)\}$

Finally, let the target ground term be  $f(c, c, c)$ . Notice that if there is a solution to the PCP then there exists a recipe term  $\zeta$  such that  $\zeta\sigma =_R f(c, c, c)$ . Furthermore, if there is a recipe term  $\zeta$  such that  $\zeta\sigma =_R f(c, c, c)$  then a solution to the PCP can be extracted from the indexes of the  $g_i$  function symbol in the term  $\zeta\sigma$ . Finally, the recipe cannot just be  $f(c, c, c)$  since  $c \in \tilde{n}$ .  $\blacktriangleleft$

► **Example 30.** Consider the following PCP:

$$\overbrace{\left(\frac{ba}{baa}\right)}^{\text{pair 1}}, \quad \overbrace{\left(\frac{ab}{ba}\right)}^{\text{pair 2}}, \quad \overbrace{\left(\frac{aaa}{aa}\right)}^{\text{pair 3}}$$

Following the construction of Lemma 29:

$$R = \left\{ \begin{array}{ll} f(b_1(a_1(x)), g_1(y), b_2(a_2(a_2(z)))) & \rightarrow f(x, y, z) \\ f(a_1(b_1(x)), g_2(y), b_2(a_2(z))) & \rightarrow f(x, y, z) \\ f(a_1(a_1(a_1(x))), g_3(y), a_2(a_2(z))) & \rightarrow f(x, y, z) \end{array} \right\}$$

And we construct the frame  $\phi = \nu\tilde{c}.\{x_1 \mapsto b_1(a_1(c)), y_1 \mapsto b_2(a_2(a_2(c))), z_1 \mapsto g_1(c), x_2 \mapsto a_1(b_1(c)), y_2 \mapsto b_2(a_2(c)), z_2 \mapsto g_2(c), x_3 \mapsto a_1(a_1(a_1(c))), y_3 \mapsto a_2(a_2(c)), z_3 \mapsto g_3(c)\}$

Then, a recipe is  $\zeta = f(b_1(a_1(x_3)), g_1(z_3), b_2(a_2(a_2(y_3))))$  since  $\zeta\sigma \rightarrow_R f(c, c, c)$ .

As a corollary of Lemma 29 we obtain the following.

► **Corollary 31.** *The deduction problem is undecidable for the class of graph-embedded convergent TRSs.*

► **Remark 32.** Note that the knowledge problems of deduction and static-equivalence were already proven undecidable in general in [1], where a reduction from PCP is also used. However, the system used in the proof from [1] is not graph-embedded and it's not clear how to directly adapt that proof to the graph-embedded case of Lemma 29.

## 4.2 Decidable Knowledge Problems

We consider below a restricted form of the graph-embedded TRS for which we can show decidability of the knowledge problems.

► **Definition 33** (Contracting TRS). *A TRS,  $R$ , is contracting if for each  $l \rightarrow r \in R$  such that  $r$  is not a constant, we have that  $l \approx r$  and  $\text{depth}(r) \leq 2$ , or  $l \rightarrow_{R_{\text{gemb}}}^+ \cdot \approx r$  and  $r$  is a well-formed term and  $\text{depth}(r) \leq 1$ , where, considering the rules of Definition 15:*

- root application of rule (1) are applied before other rules in  $R_{\text{gemb}}$ .
- if rule (1),  $f(x_1, \dots, x_n) \rightarrow x_i$ , is applied below the root then only a variable instance of it is applied and if  $x_i$  is not removed by a latter rule then there exist a rule  $l' \rightarrow x_i \in R$  and a position  $q$  such that  $l'|_q = f(x_1, \dots, x_n)$ .

## 30:12 Going Beyond Subterm Convergent Theories

- if rule (2),  $f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) \rightarrow f(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ , is applied then only a variable instance of it is applied.
- if rule (4),  $f(x_1, \dots, x_{i-1}, g(\bar{z}), x_{i+1}, \dots, x_m) \rightarrow f(x_1, \dots, x_{i-1}, \bar{z}, x_{i+1}, \dots, x_m)$ , is applied then only a variable instance of it is applied and if all  $\bar{z}$  are not removed by latter rules, then for each  $z_i$  not removed there exists a rule  $l' \rightarrow z_i \in R$  such that  $l'|_q = g(\bar{z})$ .
- for  $\approx$ , only a single  $\approx_s$ -permutation can be applied at the root followed by one or no  $\approx_1$ -permutations at the variable positions. In addition:
  - if a variable  $x$  occurs in a direct subterm  $C[x]$  of  $l$  not equal to  $x$ , and by application of  $\approx$  the variable  $x$  occurs in a direct subterm of  $r$  not equal to  $C[x]$ , then there exist a rule  $l' \rightarrow x \in R$  and a position  $q$  such that  $l'|_q = C[x]$ .

Although the definition restricts the set of graph-embedded systems it is still sufficient to model many security protocols of interest. We include several such examples below.

► **Example 34.** Consider several convergent TRSs given in previous examples:

- The theory of blind signatures from Example 2 is contracting. Consider the rule  $unblind(sign(blind(x, y), z), y) \rightarrow sign(x, z)$  and the rewriting

$$unblind(sign(blind(x, y), z), y) \rightarrow_{R_{emb}^*} sign(x, z).$$

The chain of rules in the rewriting could be:

- rule (1) - placing  $sign$  at the root,

$$unblind(sign(blind(x, y), z), y) \rightarrow_{R_{emb}} sign(blind(x, y), z),$$

- rule (4) - removing  $blind$ ,  $sign(blind(x, y), z) \rightarrow_{R_{emb}} sign(x, y, z)$ ,
- rule (2) - removing  $y$ ,  $sign(x, y, z) \rightarrow_{R_{emb}} sign(x, z)$ .

Since rule (4) was applied, removing  $blind$  and variable  $x$  was not removed by rule (2), there needs to be a rule  $l \rightarrow r$  in  $R$  such that  $x = r$ , which is the case with the rule  $unblind(blind(x, y), y) \rightarrow x$ .

- The theory of addition, introduced in Example 22, is a contracting TRS. Notice that  $plus(x, s(y)) \approx plus(s(x), y)$  and that the cap, here  $s()$ , has been removed from  $y$ . Thus, there needs to be a rule,  $l \rightarrow r$ , such that  $l|_p = s(y)$  and  $r = y$ . This is exactly the rule  $pred(s(x)) \rightarrow x$ .
- The theory of prefix with pairing from Example 23 is a contracting TRS.
- Any subterm convergent TRS such that the right-hand side is either a strict subterm or a constant is contracting, like for instance the theory of pairing with encryption,  $R = \{fst(\langle x, y \rangle) \rightarrow x, snd(\langle x, y \rangle) \rightarrow y, dec(enc(x, y), y) \rightarrow x\}$ , and the theory of Example 25.

► **Example 35.** Consider several of the previous example TRSs:

- The theory of trap-door commitment of Definition 24 is not a contracting TRS. Interestingly, this theory is also not locally stable [15]. However, if we add the rules,

$$fst(f(x_1, x_2, x_3, x_4)) \rightarrow x_1, snd(f(x_1, x_2, x_3, x_4)) \rightarrow x_2, thd(f(x_1, x_2, x_3, x_4)) \rightarrow x_3,$$

then the theory is contracting and locally stable.

- The theory of Example 20 is not contracting. Notice that for the rule  $mal(enc(x, y), z) \rightarrow enc(z, y)$ , the node labeled with  $z$  is moved under the  $enc$  node on the right-hand side. This violates the requirements of Definition 33, specifically requiring rule (3). Thus, even with additional rules, it cannot be made to be contracting. This theory is also not locally stable, as shown in [15].

► **Remark 36.** If we consider now the TRS from the undecidability proof of Lemma 29 we can see that while the system is graph-embedded it is not contracting. The system can be made contracting by adding a set of rewrite rules of the form  $a_1(x) \rightarrow x$ ,  $b_1(x) \rightarrow x$ ,  $a_2(x) \rightarrow x$ ,  $\dots$ . Then, clearly for this new system both the deduction problem and unification problem detailed in the proof are decidable.

We now develop a few results and definitions we need to show the decidability of the knowledge problems for contracting, graph-embedded convergent systems.

► **Definition 37 (Context Bound).** Let  $ar(\Sigma)$  denote the maximal arity of any function symbol in  $\Sigma$ . Define the context bound of a graph-embedded TRS,  $R = \{l_i \rightarrow r_i\}$ ,  $1 \leq i \leq n$ , as

$$c_R = \max_{1 \leq i \leq n} (|l_i|, ar(\Sigma) + 1)$$

► **Example 38.** For the theory of malleable encryption from Example 20,  $c_{R_{mal}} = 5$ . For the theory of blind signatures,  $R_{blind}$ , from Example 2,  $c_{R_{blind}} = 7$ .

► **Definition 39 (Graph-Embedded Subterms).** Let  $R$  be a contracting TRS and let  $st(t)$  be the set of subterms of a term  $t$ . Then, the set of graph-embedded subterms of a term  $t$ , denoted as  $gst(t)$ , is defined as:  $gst(c) = \{c\}$ , where  $c$  is a name or a constant,  $gst(t) = \{t' \mid t \rightarrow_{R_{gemb}}^* t' \approx t', \text{ and } t' \text{ is a well formed term}\} \cup \bigcup_{t'' \in st(t)} gst(t'')$ . Let  $\phi = \nu \tilde{n}. \sigma$  be a frame, then  $gst(\phi) = \bigcup_{t \in Ran(\sigma)} gst(t)$ .

Notice that for any term  $t$ ,  $gst(t)$  is a finite set. This is due to the fact that when recursively constructing  $gst(t)$  in the second rule of Definition 39,  $t'$  is equal or smaller in size to  $t$ , and any term  $t'' \in st(t)$  must be strictly smaller than  $t$ . Thus, we have the following result.

► **Lemma 40.** For any term  $t$  and any frame  $\phi$ ,  $gst(t)$  and  $gst(\phi)$  are finite sets.

Based on the extended definition of subterms,  $gst$ , we can now construct a saturation set for frames. Computing such a saturation set is the goal of many procedures that consider security notions such as deducibility. The saturation set represents the knowledge of the attacker and their ability to deduce a term from that knowledge, see [1] for more background.

► **Definition 41 (Frame Saturation for Contracting Convergent TRS).** Let  $\phi = \nu \tilde{n}. \sigma$  be a frame, and  $R$  a contracting convergent TRS. Define the set  $sat(\phi)$  to be the smallest set such that  $Ran(\sigma) \subseteq sat(\phi)$ , and  $n \in sat(\phi)$  for every  $n \in fn(\phi)$ , and closed under the following two rules:

1. if  $M_1, \dots, M_l \in sat(\phi)$  and  $f(M_1, \dots, M_l) \in gst(\phi)$ , then  $f(M_1, \dots, M_l) \in sat(\phi)$ ,
2. if  $M_1, \dots, M_l \in sat(\phi)$ ,  $C[M_1, \dots, M_l] \rightarrow_R^\epsilon M$ , where  $C$  is a context,  $|C| \leq c_R$ ,  $fn(C) \cap \tilde{n} = \emptyset$ , and  $M \in gst(\phi)$ , then  $M \in sat(\phi)$ .

► **Remark 42.** It is important to note that  $sat(\phi)$  should contain the set of deducible terms from the frame. For example, it would be tempting to just place all of  $gst(\phi)$  into  $sat(\phi)$  immediately, but this would add non-deducible terms to the set and invalidate the results.

Also notice for Definition 41, by applying an empty context, the second rule ensures that for any  $S \in sat(\phi)$ , if  $S \rightarrow_R^\epsilon S'$  and  $S' \in gst(\phi)$ , then  $S' \in sat(\phi)$ .

This set is also finite which is critical to computing the possible attackers knowledge thus having a finite set is useful for any practical procedure for deciding deducibility.

► **Lemma 43.** For any frame  $\phi$ ,  $sat(\phi)$  is finite.

## 30:14 Going Beyond Subterm Convergent Theories

**Proof.** New terms not originally contained in  $\phi$  are only added to  $\text{sat}(\phi)$  if they are first contained in  $\text{gst}(\phi)$ . Since  $\text{gst}(\phi)$  is finite by Lemma 40,  $\text{sat}(\phi)$  is finite. ◀

The following definition and lemma will be useful in proving the main motivating result as they show key components of the local stability property given in Definition 46.

► **Definition 44** (Closure Under Small Context). *Let  $\phi = v\tilde{n}.\sigma$  be a frame, and  $R$  a convergent TRS. A finite set of ground terms,  $\mathcal{S}$ , is closed under small  $\phi$ -restricted context by  $R$  if the following property holds: for any context  $C$  with  $|C| \leq c_R$  and  $\text{fn}(C) \cap \tilde{n} = \emptyset$ , and any  $S_1, \dots, S_l \in \mathcal{S}$ , if  $C[S_1, \dots, S_l] \rightarrow_R^\epsilon M$  then there exist a context  $C'$  and  $S'_1, \dots, S'_k \in \mathcal{S}$  such that  $|C'| \leq c_R^2$ ,  $\text{fn}(C') \cap \tilde{n} = \emptyset$ , and  $M \rightarrow_R^* C'[S'_1, \dots, S'_k]$ . When  $\phi$  is clear from the context,  $\mathcal{S}$  is said to be closed under small context by  $R$ .*

Using  $c_R^2$  as an upper bound is somewhat arbitrary since we need just some fixed bound. We use  $c_R^2$  since it is sufficient for the results in this paper and it is the bound used in [1].

► **Lemma 45.** *For any frame  $\phi$  and any contracting convergent TRS  $R$ , let  $\text{sat}(\phi)$  be the set given in Definition 41. Then,  $\text{sat}(\phi)$  is closed under small context by  $R$ .*

**Proof.** See Appendix A. ◀

We can now introduce the local stability property, which if satisfied ensures the decidability of deduction. The local stability property was introduced in [1] and improved in [6]. A simplified version of this definition is introduced below. It is simplified because we don't consider  $AC$ -symbols as in [1, 6].

► **Definition 46** (Local Stability [1]). *A convergent TRS,  $R$ , is locally stable if, for every frame  $\phi = v\tilde{n}.\sigma$ , where  $\sigma$  is a ground  $R$ -normalized substitution, there exists a finite set  $\text{sat}(\phi)$  of ground terms such that:*

- $\text{Ran}(\sigma) \subseteq \text{sat}(\phi)$  and  $n \in \text{sat}(\phi)$ , for all  $n \in \text{fn}(\phi)$ ;
- if  $M_1, \dots, M_k \in \text{sat}(\phi)$  and  $f(M_1, \dots, M_k) \in \text{st}(\text{sat}(\phi))$ , then  $f(M_1, \dots, M_k) \in \text{sat}(\phi)$ ;
- if  $C[S_1, \dots, S_l] \rightarrow_R^\epsilon M$ , where  $C$  is a context with  $|C| \leq c_R$  and  $\text{fn}(C) \cap \tilde{n} = \emptyset$ , and  $S_1, \dots, S_l \in \text{sat}(\phi)$ , then there exist a context  $C'$  and  $S'_1, \dots, S'_k \in \text{sat}(\phi)$  such that  $|C'| \leq c_R^2$ ,  $\text{fn}(C') \cap \tilde{n} = \emptyset$ , and  $M \rightarrow_R^* C'[S'_1, \dots, S'_k]$ ;
- if  $M \in \text{sat}(\phi)$  then  $\phi \vdash M$ .

► **Remark 47.** The existence of a set  $\text{sat}(\phi)$  in the above definition means that any set satisfying the conditions of Definition 46 is sufficient. In Definition 41 we give a particular such set for contracting convergent TRSs which satisfies the conditions of Definition 46, as shown below.

In a locally stable TRS, any deduction problem reduces to check finitely many terms that could be possible recipes of the input, and any static equivalence problem reduces to checking finitely many equations between bounded terms satisfying the name restriction [1].

For any contracting convergent TRS, Lemma 45 establishes all but the last item of Definition 46, and this item has already been shown in [1].

► **Lemma 48** ([1]). *For any frame  $\phi$  and any ground term  $M$ , if  $M \in \text{sat}(\phi)$  then  $\phi \vdash M$ .*

This result is proven in [1] where they also consider the more complicated case of systems with  $AC$ -symbols.

► **Theorem 49.** *Any contracting convergent TRS is locally stable.*



**Proof.** The first two conditions follow from Definition 41 where  $\text{sat}(\phi)$  is given in the particular case of a contracting convergent TRS. Then, the third condition follows from Lemma 45. The final condition follows from Lemma 48. ◀

► **Example 50.** Continuing Example 34:

- Since the theory of blind signatures is a contracting convergent TRS, it is locally stable by Theorem 49. Since this theory doesn't contain an  $AC$ -symbol it is also locally finite [1]. The theory of blind signatures being both locally stable and locally finite, both deduction and static-equivalence are decidable [1].
- All the theories from Example 34 are thus locally stable and locally finite, thus both knowledge problems are decidable.
- By the same argument the theory from Example 22 is locally stable and both knowledge problems are decidable.
- By the same argument any subterm convergent theory such that the right-hand side is either a strict subterm or a constant is locally stable and both knowledge problems are decidable.

Directly from Theorem 49 and the result in [1], which establish the decidability of deduction and static equivalence for locally stable and finite theories, we obtain the following corollary.

► **Corollary 51.** *The deduction and static equivalence problems are both decidable for the class of contracting convergent TRSs.*

## 5 Relation to the Finite Variant Property

The Finite Variant Property (FVP) is a useful property which is utilized in a number of applications, including protocol analysis. See for example [12, 17]. Before discussing the relation it's useful to introduce the following definition.

► **Definition 52 (Boundedness Property).** *A convergent TRS,  $R$ , has the boundedness property if  $\forall t \exists n \forall \sigma : t(\sigma \downarrow) \rightarrow_{\bar{R}}^{\leq n} (t\sigma) \downarrow$ . That is, for any term  $t$  there exists a bound,  $n$ , on the number of step required to reach the normal form, and this bound is independent of the substitution.*

► **Remark 53.** It's been shown in [11] that a TRS has the FVP iff it has the boundedness property of Definition 52. See also [13, 18] for more background.

One could naturally ask if the graph-embedded or contracting definitions just lead to systems with the FVP. This is not the case but some of the examples above, such as blind signatures, do have the FVP. This is not surprising, given that the FVP can be useful for showing things like termination. Therefore, a more interesting question could be: are there interesting examples from the protocol analysis literature for which deduction and static equivalence are decidable, do not have the FVP, but are representable by contracting convergent TRSs? Here we answer positively this question.

► **Example 54.** Consider again the theory of Addition,  $R_{add}$ , from Example 22.  $R_{add}$  is a contracting convergent TRS, is locally stable, and contains no  $AC$ -symbols, thus deduction and static equivalence are decidable. However,  $R_{add}$  does not have the FVP, we can see this by considering the rule  $\text{plus}(x, s(y)) \rightarrow \text{plus}(s(x), y)$  and the boundedness property. Notice that for any finite bound  $n$  one can select a normal form substitution,  $\sigma$ , such that  $\text{plus}(x, s(y))\sigma \rightarrow_{\bar{R}_{add}}^{\geq n} (\text{plus}(x, s(y))\sigma) \downarrow$ . Namely,  $\sigma = \{y \mapsto s^{n+1}(z)\}$ . Since  $R_{add}$  does not have the boundedness property it can't have the FVP [11]. Yet,  $R_{add}$  is a contracting convergent TRS. Notice that the second and third rules are already subterm. The first rule is obtained by applying Definition 13. Therefore,  $R_{add}$  satisfies Corollary 51.

## 6 Conclusions and Future Work

In this paper, we have introduced the idea of graph-embedded term rewrite systems and shown their applicability in protocol analysis for identifying protocols with the local stability property. This in turn allows for the identification of protocols with decidable deduction and static-equivalence problems. However, this could be just the first step as there are many additional questions about the use of graph-embedded systems applied to the protocol analysis domain and also outside that domain.

With respect to the current paper, a natural question arises. While the knowledge problems are undecidable for graph-embedded convergent systems in general and that they are decidable for contracting, graph-embedded convergent systems, there is a gap between the two classes of systems. That is, how much can the contracting subclass be extended before the undecidable barrier is encountered? In this direction, we could try to weaken the rule application strategy used in Definition 33.

With respect to additional security protocol applications there are several interesting areas that could be explored:

- It would be interesting to consider termination conditions of various procedures [1, 12, 15, 17] with respect to graph-embedded systems. That is, do graph-embedded systems provide any help with obtaining termination guarantees?
- The cap problem, developed in [5] could be viewed as a particular form of deduction where one wants to determine whether a given secret constant is deducible or not. We conjecture that the cap problem is decidable for contracting convergent TRSs and undecidable for general graph-embedded convergent TRSs. A proof of this would be useful.
- It would be useful to consider additional properties that have been developed for use in protocol analysis. For example, *layer-convergence* is a property developed in [9] where it's shown that the YAPA procedure for protocol analysis will not fail on theories with this property. While termination is not ensured, this does provide a useful condition for knowing if the procedure can be used. Currently, we don't know how the graph-embedded property compares to layer-convergence. Note, checking for the graph-embedded property is relatively easy. Thus, if a restricted form of graph embedding could be shown to be related to layer-convergence, then such a graph embedding could be a useful way to identify layer-convergent systems. This in turn would be useful for identifying systems for which the YAPA procedure could be applied.

With respect to graph theory ideas, we are also interested in knowing if additional graph theory ideas could be useful in symbolic security protocol analysis:

- Of course not absolutely all theories considered in [1, 12, 15, 17] are graph-embedded. It would be interesting to know if such systems could be considered via graph minor concepts?
- In addition, are graph minor relations such as topological minors useful? Given standard graph theory this would seem not to be the case. The containment of the topological minor relation in the graph minor relation would appear to rule this out. However, this may not be completely true in the term graph domain, where we must obey the standard well-formed term requirements.

---

## References

- 1 Martín Abadi and Véronique Cortier. Deciding knowledge in security protocols under equational theories. *Theor. Comput. Sci.*, 367(1-2):2–32, 2006.

- 2 Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In Chris Hankin and Dave Schmidt, editors, *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK, January 17-19, 2001*, pages 104–115. ACM, 2001.
- 3 Siva Anantharaman, Hai Lin, Christopher Lynch, Paliath Narendran, and Michaël Rusinowitch. Unification modulo homomorphic encryption. *J. Autom. Reason.*, 48(2):135–158, 2012.
- 4 Siva Anantharaman, Paliath Narendran, and Michael Rusinowitch. Intruders with caps. Research report, Laboratoire d’Informatique Fondamentale d’Orléans, 2007. URL: <https://hal.science/hal-00144178>.
- 5 Siva Anantharaman, Paliath Narendran, and Michaël Rusinowitch. Intruders with caps. In Franz Baader, editor, *Term Rewriting and Applications, 18th International Conference, RTA 2007, Paris, France, June 26-28, 2007, Proceedings*, volume 4533 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2007.
- 6 Mauricio Ayala-Rincón, Maribel Fernández, and Daniele Nantes-Sobrinho. Intruder deduction problem for locally stable theories with normal forms and inverses. *Theor. Comput. Sci.*, 672:64–100, 2017.
- 7 Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.
- 8 Franz Baader and Wayne Snyder. Unification theory. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 445–532. Elsevier and MIT Press, 2001.
- 9 Mathieu Baudet, Véronique Cortier, and Stéphanie Delaune. YAPA: A generic tool for computing intruder knowledge. *ACM Trans. Comput. Log.*, 14(1):4, 2013.
- 10 Bruno Blanchet. Automatic proof of strong secrecy for security protocols. In *2004 IEEE Symposium on Security and Privacy (S&P 2004), 9-12 May 2004, Berkeley, CA, USA*, pages 86–100. IEEE Computer Society, 2004.
- 11 Christopher Bouchard, Kimberly A. Gero, Christopher Lynch, and Paliath Narendran. On forward closure and the finite variant property. In Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt, editors, *Frontiers of Combining Systems - 9th International Symposium, FroCoS 2013, Nancy, France, September 18-20, 2013. Proceedings*, volume 8152 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2013.
- 12 Rohit Chadha, Vincent Cheval, Ștefan Ciobăcă, and Steve Kremer. Automated verification of equivalence properties of cryptographic protocols. *ACM Trans. Comput. Log.*, 17(4):23:1–23:32, 2016.
- 13 Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property: How to get rid of some algebraic properties. In Jürgen Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2005.
- 14 Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *J. Comput. Secur.*, 14(1):1–43, 2006.
- 15 Ștefan Ciobăcă, Stéphanie Delaune, and Steve Kremer. Computing knowledge in security protocols under convergent equational theories. *J. Autom. Reasoning*, 48(2):219–262, 2012.
- 16 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, third edition, 2006.
- 17 Jannik Dreier, Charles Duménil, Steve Kremer, and Ralf Sasse. Beyond subterm-convergent equational theories in automated verification of stateful protocols. In Matteo Maffei and Mark Ryan, editors, *Principles of Security and Trust - 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10204 of *Lecture Notes in Computer Science*, pages 117–140. Springer, 2017.
- 18 Santiago Escobar, Ralf Sasse, and José Meseguer. Folding variant narrowing and optimal variant termination. *J. Log. Algebr. Program.*, 81(7-8):898–928, 2012.

## A Technical Appendix

Let us first introduce some additional notations that are useful in the proofs of our results. Given a set  $\bar{S}$  of ground term, a frame  $\phi = \nu\tilde{n}.\sigma$  and a TRS  $R$ , a *context instantiated with terms in  $\bar{S}$*  is a ground term of the form  $u\alpha$  where  $u$  is a term including no names in  $\tilde{n}$ , and  $\alpha$  a substitution such that  $\text{Dom}(\alpha) = \text{Var}(u)$  and  $\text{Ran}(\alpha) \subseteq \bar{S}$ . The term  $u$  is called the *context part of  $t$* . The term  $t$  is called a *small context* if the size of its context part  $u$  satisfies  $|u| \leq c_R$ . A term denoted by  $u[S_1, \dots, S_n]$  for  $S_1, \dots, S_n \in \bar{S}$  corresponds to the context instantiated with terms in  $\bar{S}$  defined by  $u[x_1, \dots, x_n]\{x_1 \mapsto S_1, \dots, x_n \mapsto S_n\}$ , where the set of variables in  $u[x_1, \dots, x_n]$  is assumed to be  $\{x_1, \dots, x_n\}$ .

### A.1 Additional Lemmas

The following two technical lemmas are useful to prove the closure property stated by Lemma 45 for any contracting convergent TRS.

A rewrite step applied at the root position is denoted by  $\xrightarrow{\epsilon}$ , while a rewrite step applied at some non-rooted position is denoted by  $\xrightarrow{\neq\epsilon}$ .

► **Lemma 55.** *Let  $R$  be a contracting convergent TRS. Let  $l \xrightarrow{\neq\epsilon^+}_{R_{\text{gemb}}} \cdot \approx r$  such that  $l$  is a subterm of the left-hand side  $l'$  of a rule  $l' \rightarrow r$  in  $R$  where  $\text{depth}(r) = 1$ . For any variable  $x$  in  $r$  occurring in  $l$  at a position  $p$  such that  $|p| > 1$ , and any substitution  $\varphi$ , the following is true: for any positions  $q, q'$  such that  $\epsilon < q < q' \leq p$ ,  $q' = q.i$ ,  $l(q)$  is a function symbol  $f$ , if  $l|_q\varphi \in \text{sat}(\phi)$  then there exists a projecting rule  $C[f(x_1, \dots, x_n)] \rightarrow x_i$  in  $R$ .*

**Proof.** The proof is by induction on  $dp(l) = \max_{\{p \mid l(p) \in \text{Var}(r)\}} |p|$ . For the base case, one can check that the property holds for  $dp(l) = 2$  due the particular form of  $r$ . For the induction step, consider a derivation

$$\begin{aligned} l[f(y_1, \dots, y_{j-1}, g(\dots, x_i, \dots), y_{j+1}, \dots, y_n)] \\ \xrightarrow{\neq\epsilon^+}_{R_{\text{gemb}}} l[f(v_1, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_n)] \\ \xrightarrow{\neq\epsilon^*}_{R_{\text{gemb}}} r \end{aligned}$$

where:

- the  $\xrightarrow{\neq\epsilon^+}_{R_{\text{gemb}}}$  derivation consists of a single rule (1) or a single rule (4) followed by the repeated application of rule (2) to retrieve a well-formed term of arity  $n$ ;
- $x_i$  occurs in  $l$  at a position  $p$  such that  $|p| = dp(l)$  and  $x_i \in \text{Var}(r)$ .

Let  $l' = l[f(v_1, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_n)]$ . We have  $dp(l') < dp(l)$  and so we can assume that the property holds for  $l'$ . Moreover,  $v_j$  occurs in  $r$ , otherwise it would contradict that  $x_i$  occurs in  $r$ . By the induction hypothesis, there must exist a projecting rule to get  $v_j$  from  $f(v_1, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_n)$ . This projection rule can be reused for the redex  $f(y_1, \dots, y_{j-1}, g(\dots, x_i, \dots), y_{j+1}, \dots, y_n)$  of  $l$ . By Definition 33, there exists a projecting rule to get  $x_i$  from the subterm  $g(\dots, x_i, \dots)$  of  $l$ . For all the strict subterms of  $l$  above the redex which are on the path to the root of  $l$ , we can reuse all the projecting rules available for all the strict subterms of  $l'$  above  $f(v_1, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_n)$  which are on the path to the root of  $l'$ . Consequently, there is a projecting rule for all the strict subterms of  $l$  above  $x_i$  at position  $p$  which are on the path to the root of  $l$ . Thus, the property holds for  $l$ . ◀

► **Lemma 56.** *Let  $l \approx r$  be a rule of a contracting convergent TRS such that  $\text{depth}(l) = \text{depth}(r) = 2$ . For any variable  $x$  in  $r$  occurring at a position  $p$  of  $l$  in a non-variable direct subterm  $l|_q$  of  $l$  which is not a direct subterm of  $r$ , and any substitution  $\varphi$ , the following is true: if  $l|_q\varphi \in \text{sat}(\phi)$ , then  $l|_p\varphi \in \text{sat}(\phi)$ .*

**Proof.** Thanks to Definition 33. ◀

## A.2 Proof of Lemma 45

**Proof.** Let us analyse the different forms of derivation that may occur in the definition a rule of a contracting TRS.

First, if  $l \xrightarrow{\epsilon}_{R_{\text{geom}}} r$ , then  $r$  is direct subterm of  $l$ . According to [1], for any term  $t$  corresponding to a small context instantiated by terms in  $\text{sat}(\phi)$  such that  $t = l\varphi$ , the term  $r\varphi$  remains a small context instantiated by terms in  $\text{sat}(\phi)$ . Then, we prove by induction on the length of the derivation that the same property also holds for  $\xrightarrow{\epsilon^*}_{R_{\text{geom}}}$ .

Second, consider a derivation  $l \xrightarrow{\neq\epsilon^+}_{R_{\text{geom}}} r$  where  $\text{depth}(r) \leq 1$ . The case  $\text{depth}(r) = 0$  is easy since it corresponds to the classical subterm case (see above). So, let us assume  $r = h(x_1, \dots, x_n)$  and a small context instantiated by terms in  $\text{sat}(\phi)$ , say  $t$ , such that  $t = l\varphi$ . Let  $i$  be any integer in  $\{1, \dots, n\}$ . Any  $x_i$  occurs in  $l$  either at depth at most 1 or at some depth strictly greater than 1.

- If  $x_i$  occurs in  $l$  at depth at most 1, then  $x_i\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$ .
- if  $x_i$  occurs in  $l$  at some depth strictly greater than 1, then  $x_i\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$  by Lemma 55.

Then,  $r\varphi = h(x_1\sigma, \dots, x_n\sigma)$  is a small context instantiated by terms in  $\text{sat}(\phi)$ . Indeed, by construction, the context part of  $r\varphi$  cannot be greater than the context part of  $l\sigma$ .

Third, consider the case  $l \approx r$  where  $\text{depth}(r) \leq 2$ . By definition of  $\approx$ , we have  $\text{depth}(l) = \text{depth}(r)$ . The case  $\text{depth}(r) \leq 1$  being easy, let us assume  $\text{depth}(l) = \text{depth}(r) = 2$ ,  $r = h(r_1, \dots, r_n)$ , and a small context instantiated by  $\text{sat}(\phi)$ , say  $t$ , such that  $t = l\varphi$ . Let  $i$  be any integer in  $\{1, \dots, n\}$ .

- If  $r_i$  is a variable occurring at depth at most 1 in  $l$ , then  $r_i\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$ .
- If  $r_i$  is a variable occurring at depth 2 in  $l$ , then  $r_i\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$  by Lemma 56.
- If  $r_i$  is a non-variable term occurring as a direct subterm  $l_j$  of  $l$  for some  $j \in [1, n]$ , then  $r_i\varphi = l_j\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$ .
- If  $r_i$  is a non-variable term  $f(\bar{x})$  not occurring as a direct subterm of  $l$ , then there are two cases for any variable  $x \in \bar{x}$ : if  $x$  occurs at depth at most 1 in  $l$ , then  $x\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$ ; otherwise  $x$  also occurs at depth 2 in  $l$  and  $x\varphi$  is a small context instantiated by terms in  $\text{sat}(\phi)$  by Lemma 56.

Then,  $r\varphi = h(r_1\sigma, \dots, r_n\sigma)$  is a small context instantiated by terms in  $\text{sat}(\phi)$ . Indeed, by definition of  $\approx$ , the context part of  $r\varphi$  cannot be greater than the context part of  $l\sigma$ . ◀