Computing finite variants for subterm convergent rewrite systems [⋆]

Ştefan Ciobâcă

LSV, ENS Cachan & CNRS

Abstract. Driven by an application in the verification of security protocols, we introduce the strong finite variant property, an extention of the finite variant property defined in [1] and we show that subterm convergent rewrite systems enjoy the strong finite variant property modulo the empty equational theory.
We argue that the strong finite variant property is more natural and more useful in practice than the finite variant property. We also compare the two properties and we provide a prototype implementation of an algorithm that computes a finite strongly complete set of variants for any term t with respect to a subterm convergent rewrite system.

1 Introduction

Given a term (e.g. $t = \mathbf{dec}(x, y)$) and a convergent term rewriting system (e.g. $\mathcal{R} = \{\mathbf{dec}(\mathbf{enc}(x, y), y) \to x\}$), we are interested in having a convenient symbolic representation of all normal forms $t\sigma{\downarrow}$ of instantiations $t\sigma$ of the term t.

In the above case, the normal form of $t\sigma$ will fall into one of the following two cases:

1. either $\sigma(x) =_{\mathcal{R}} \mathbf{enc}(s, \sigma(y))$ for some term s, in which case $t\sigma{\downarrow} = s{\downarrow}$
2. or $\sigma(x) \neq_{\mathcal{R}} \mathbf{enc}(s, \sigma(y))$ for any term s, in which case $t\sigma{\downarrow} = t(\sigma{\downarrow})$ (where $\sigma{\downarrow}$ denotes the normal form of σ).

Informally, rewrite systems for which instantiations of any term t can be classified into a finite number of categories such as above are said to have the *finite variant property.*

The finite variant property is useful in symbolic analysis of security protocols [1] and in solving unification and disunification problems [1,2].

Contributions. We work with subterm convergent rewrite systems, a class of rewrite systems relevant to security protocol analysis [3]. We show that these rewrite systems have the finite variant property and a slightly stronger property which we call the *strong finite variant property.* The proofs are constructive and we implement the algorithm for computing a strongly complete finite set of variants of a term in the tool SubVariant.

[⋆] This work has been partly supported by the ANR SeSur AVOTÉ.

Related work. The finite variant property was first introduced in [1]. In [4], sufficient conditions and necessary conditions for the finite variant property are introduced. Variant narrowing [2] is a complete procedure for equational unification inspired by the finite variant property. A modular proof method for termination based on the notion of variant is proposed in [5]. Several techniques [6,7,8] for verifying security protocols make use of the finite variant property as a substep in the algorithm.

2 Preliminaries

We consider standard notations for a term algebra: a finite *signature* $\mathcal{F}$, each function symbol $f \in \mathcal{F}$ having an arity $\mathsf{ar}(f) \in \mathbb{N}$, a countably infinite set of $\mathcal{X}$ of *variables*, the set $\mathcal{T}(\mathcal{X}_0)$ denoting all *terms* bulid inductively from the variables $\mathcal{X}_0 \subseteq \mathcal{X}$ by applying function symbols from $\mathcal{F}$. Given a term t, $\mathsf{vars}(t)$ is the set of variables appearing in t.

Substitutions are defined as usual, with $t\sigma$ denoting the term t after application of the substitution σ. The identity substitution is denoted id. The restriction of a substitution σ to a set X of variables is denoted $\sigma[X]$.

We define *positions* as usual, with $\mathsf{pos}(t)$ denoting the positions of a term $t \in \mathcal{T}(\mathcal{X})$. We denote the subterm of t at position $p \in \mathsf{pos}(t)$ by $t|_p$. The term t with position $p \in \mathsf{pos}(t)$ instantiated to s is denoted $t[s]_p$. If $t \in \mathcal{T}(\mathcal{X})$, we will denote by $\mathsf{st}(t)$ the set of *subterms* of t. If $s \in \mathsf{st}(t)$, we write $s \sqsubseteq t$. By $\mathsf{mgu}(s,t)$ we denote the most general unifier of two unifiable terms s and t.

If $\mathcal{R}$ is a rewrite system, we use $\to_{\mathcal{R}}$ for the one-step rewrite relation defined by R and $\to_{\mathcal{R}}^*$ for the transitive and reflexive closure of $\to_{\mathcal{R}}$. If $\mathcal{R}$ is convergent, we will denote by $t{\downarrow_{\mathcal{R}}}$ the normal form of t. We say that two terms s and t are *equal modulo* $\mathcal{R}$, and we write $s =_{\mathcal{R}} t$, if $s{\downarrow} = t{\downarrow}$.

We are interested in a particular class of convergent term rewriting systems:

Definition 1 (subterm convergent rewrite system).
A rewrite system $\mathcal{R}$ is called subterm convergent *if it is convergent and if for all rewrite rules $l \to r$ of $\mathcal{R}$ we have:*

1. *either $r \sqsubseteq l$ (we then call $l \to r$ a* subterm rule*)*
2. *or $\mathsf{vars}(r) = \emptyset$ and $r = r{\downarrow_{\mathcal{R}}}$ (we then call $l \to r$ an* extended rule*)*

The first type of rewrite rule, which justifies the name of these rewrite systems, was introduced in [3]. Subsequently, in [9], the extended rules were introduced. We treat here both types of rules.

3 The finite variant property

For any convergent term rewriting system $\mathcal{R}$ and any term t, we define the notion of complete set of variants of t with respect to $\mathcal{R}$:

Definition 2. *A set of substitutions $\{\sigma_1, \ldots, \sigma_n\}$ is a* complete set of variants *of a term t (with respect to $\mathcal{R}$) if for any substitution ω, we have that $(t\omega){\downarrow} = (t\sigma_i){\downarrow}\tau$ for some $1 \leq i \leq n$ and some substitution τ.*

Note that the difficulty in the above definition is that the term $(t\sigma_i){\downarrow}\tau$ is not normalized after the application of the substitution τ to the term $(t\sigma_i){\downarrow}$. Therefore all the rewrite steps to reach a normal form from $t\omega$ must be captured by some substitution σ_i as it is demonstrated bellow in Example 1. This means in particular that the set $\{\mathsf{id}\}$ consisting of the identity substitution is in general not a finite complete set of variants.

A convergent term rewriting system $\mathcal{R}$ is said to have the finite variant property if any term t admits a finite complete set of unifiers with respect to the rewrite system $\mathcal{R}$.

Example 1. Let $t = \mathbf{dec}(x, y)$ and $\mathcal{R} = \{\mathbf{dec}(\mathbf{enc}(x, y), y) \to x\}$. Then we have that $\sigma_1 = \mathsf{id}$ (the identity substitution) and $\sigma_2 = \{x \mapsto \mathbf{enc}(z, y)\}$ form a complete set of variants of t.

Indeed, for any substitution ω in normal form, we have that $\mathbf{dec}(x, y)\omega{\downarrow} = \mathbf{dec}(x, y)\omega$ (if the decryption does not succeed at the head) or $\mathbf{dec}(x, y)\omega{\downarrow} = t'$ if the decryption succeeds and therefore $x\omega = \mathbf{enc}(t', y\omega)$.

The following example illustrates that a finite complete set of variants does not always exist.

Example 2. We consider the term rewriting system $\mathcal{R} = \{f(g(x)) \to g(f(x))\}$ and the term $t = f(x)$. By analyzing the substitutions $\omega_i = \{x \mapsto g^i(y)\}$ $(i \in \mathbb{N})$ and the normal forms $t\omega_i{\downarrow} = g^i(f(y))$ $(i \in \mathbb{N})$, it can be proven that any complete set of variants of t will contain all of the substitutions $\sigma_i = \{x \mapsto g^i(y)\}$ for $i \in \mathbb{N}$ and up to renaming of the variable y. Therefore this term rewriting system does not have the finite variant property.

Rewrite systems for which any term t has a finite complete set of variants are said to have the *finite variant property.*

4 The strong finite variant property

We now define what is a strongly complete set of variants of a term t with respect to a convergent term rewriting system $\mathcal{R}$:

Definition 3. *A set of substitutions $\sigma_1, \ldots, \sigma_n$ is a* strongly complete set of variants *of t (with respect to the rewrite system $\mathcal{R}$) if for any substitution ω, we have that $\omega[X]{\downarrow} = (\sigma_i{\downarrow}\tau)[X]$[1] for some substitution τ and some σ_i such that $(t\omega){\downarrow} = (t\sigma_i){\downarrow}\tau$, where $X = \mathsf{vars}(t)$ is the set of variables appearing in t.*

[1] Recall that the notation $\omega[X]$ denotes the restriction of the substitution ω to the variables in X.

Note that in the above definition the condition $\omega[X]\!\downarrow = (\sigma_i\!\downarrow\!\tau)[X]$ does not in general imply $(t\omega)\!\downarrow = (t\sigma_i)\!\downarrow\!\tau$: take $\mathcal{R} = \{\mathbf{dec}(\mathbf{enc}(x, y), y) \to x\}$, $t = \mathbf{dec}(x, y)$, $\omega = \mathbf{enc}(z, y)$, $\sigma_i = \mathsf{id}$ (the identity substitution). We have that $\tau = \omega$ is such that $\omega[X]\!\downarrow = (\sigma_i\!\downarrow\!\tau)[X] = \tau[X]$ but $(t\omega)\!\downarrow = z \neq (t\sigma_i)\!\downarrow\!\tau = \mathbf{dec}(\mathbf{enc}(z, y), y)$.

A convergent term rewriting system $\mathcal{R}$ is said to have the strong variant property if any term t admits a finite strongly complete set of variants.

As with complete sets of variants, a finite strongly complete set of variants does not exist in general.

The main difference is that in the strong version, we ask that the substitutions σ_i match the normal forms of all variables appearing in t. The following example illustrates this idea and shows that the notion of *complete set of variants* and the notion of *strongly complete set of variants* do not coincide.

Example 3. We consider the (subterm convergent) term rewriting system

$$\mathcal{R} = \{h(f(x), y) \to y, h(g(x), y) \to y\}$$

and the term $t = h(x, y)$.

The $S = \{\sigma_1 = \mathsf{id}, \sigma_2 = \{x \mapsto f(z)\}$ is a complete finite set of variants of t. Note that S does not contain the substitution $\sigma_3 = \{x \mapsto g(z)\}$.

However, S is not a strongly complete set of variants of t: if we consider the substitution $\omega = \{x \to g(a)\}$ for some constant a, we have that:

1. $\omega\!\downarrow = \sigma_1\tau_1$ (with $\tau_1 = \{x \mapsto g(a)\}$), but $t\omega\!\downarrow \neq t\sigma_1\!\downarrow\!\tau_1$.
2. $\omega\!\downarrow \neq \sigma_2\tau_2$ for any substitution τ_2.

However, the set $S \cup \{\sigma_3\}$ is a strongly complete set of variants of t.

One application of the finite variant property is in solving unification problems $s \overset{?}{=}_{\mathcal{R}} t$ modulo the rewrite system by treating the equality sign as a free function symbol and then finding all variants of the equation. In this context of equational unification, we argue that strongly complete sets of variants are more natural:

Example 4. Continuing Example 3, if only complete sets of variants (and not strongly complete sets) are used for equational unification, some unifiers are missed. The equation

$$h(z, y) \overset{?}{=}_{\mathcal{R}} y$$

has S (defined in Example 3) as a complete set of variants. Starting from S, only the unifier $\{z \mapsto f(x)\}$ is found. However, by considering the strongly complete set of variants $S \cup \{\sigma_3\}$, the unifier $\{z \mapsto g(x)\}$ is found as well.

Another application is the verification of security protocols where strongly complete set of variants can be used to get rid of the equational theory.

4.1 Strict containment

It is easy to see that a term rewriting system having the strong finite variant property also has the finite variant property. The reverse is not true: term rewriting systems having the finite variant property need not have the strong finite variant property. Let us consider the signature $\mathcal{F} = \{f/1, g/1, c_0/0, c_1/0, \ldots\}$ and the following convergent term rewriting system

$$\mathcal{R} = \{f(g(x)) \to f(x)\}.$$

It is easy to observe that any term t has a normal form which is either $t\downarrow = g^n(f^m(x))$ or $t\downarrow = g^n(f^m(c_k))$ for some variable x and some integers n, m, k.

The identity substitution id forms by itself a complete set of variants of any term t built over the signature $\mathcal{F}$.

However, $\mathcal{R}$ does not have the strong finite variant property. This is illustrated by the following example.

Example 5. Let $t = f(x)$. By analyzing the instantiations $t\omega_i$ where the substitutions ω_i are defined $\omega_i = \{x \mapsto g^i(y)\}$ $(i \in \mathbb{N})$, it can be shown that $\sigma_i[\{x\}] = \{x \mapsto g^i(z)\}$ $(i \in \mathbb{N})$ must be contained in any strongly complete set of variants (up to renaming of z). Therefore any strongly complete set of variants of t is infinite.

4.2 In the presence of free symbols

The above example depends on the signature $\mathcal{F}$. Indeed, in the presence of a free symbol of arity greater than or equal to 2, we have that the two notions coincide since a finite complete set of variants of the term $\mathit{tuple}(t, x_1, \ldots, x_n)$ (where $\mathsf{vars}(t) = \{x_1, \ldots, x_n\}$ and where *tuple* is a free function symbol) is a strongly complete set of variants of the term t.

Note that the free symbol *tuple* of arity $n + 1$ can be *encoded* by a free symbol of arity ≥ 2 by replacing, for example, the term $\mathit{tuple}(t_1, \ldots, t_k)$ with $f(t_1, f(t_2, f(\ldots, f(t_{k-1}, t_k))))$ in case f is a free binary symbol.

We have shown that the notions of strong finite variant property and finite variant property coincide when the signature contains a free function symbol of arity ≥ 2. This follows because a complete set of variants of $\mathit{tuple}(t, x_1, \ldots, x_n)$ is a strongly complete set of variants of t. However note that even in the presence of such a free symbol, a *complete* set of variants of t is not always *strongly complete* for t (see Example 3).

5 Algorithm for a (strongly) complete set of finite variants

We show that subterm convergent term rewriting systems have the strong finite variant property by giving an algorithm that computes a finite strongly complete set of variants for a term t and for a subterm convergent rewrite system $\mathcal{R}$.

In the following we denote by $p\uparrow$ the set of all positions that are descendants of p (including p itself): $p\uparrow = \{q \mid q = p \cdot p' \textit{ for some } p'\}$.

The algorithm we present for computing a strongly complete finite set of variants is based on a refinement of *narrowing*. Each narrowing step (denoted hereafter $\hookrightarrow$) works on a configuration $(t, \mathcal{P}, \sigma)$ consisting of a term t, a set of positions $\mathcal{P}$ of t at which we will apply narrowing and a substitution σ in which a variant will be accumulated.

$$\frac{\begin{array}{ll} p \in \mathcal{P} & \\ l \to r \in \mathcal{R} & \mathsf{vars}(\{l, r\}) \cap \mathsf{vars}(t) = \emptyset \\ \theta = \mathsf{mgu}(l, t|_p) & \end{array}}{(t, \mathcal{P}, \sigma) \hookrightarrow (t\theta[r\theta]_p, \mathcal{P} \setminus p\uparrow, \sigma \circ \theta)}$$

Fig. 1. Narrowing step

To compute a complete finite set of variants of some term t, we will begin with the initial configuration $\mathcal{C}_0 = (t, \mathsf{pos}_{init}(t), \mathsf{id})$ where $\mathsf{pos}_{init}(t)$ denotes all non-variable positions of t and non-deterministically apply narrowing steps.

Each narrowing step non-deterministically chooses a rewrite rule $l \to r$ and a position p from $\mathcal{P}$ where narrowing is performed. The choice of $\mathcal{P} = \mathsf{pos}_{init}(t)$ in the initial configuration is a way to enforce the *basic restriction*, that is, narrowing is only performed strictly inside t (and not inside the variables of t). Furthermore, if we have performed narrowing at a position p and because of the specificity of subterm convergent rewrite systems, there is no need to consider this position or any of its descendants anymore and therefore they are removed from $\mathcal{P}$. At each narrowing step, the variant of the initial term is accumulated in σ. If by $\hookrightarrow^*$ we denote the reflexive-transitive closure of $\hookrightarrow$, we have that:

Theorem 1 (Correctness).
If $\mathcal{R}$ is a subterm convergent rewrite system, then the set

$$\Sigma = \{\sigma \mid (t, \mathsf{pos}_{init}(t), \mathsf{id}) \hookrightarrow^* (t', \mathcal{P}', \sigma)\}$$

is a finite complete set of variants of t with respect to $\mathcal{R}$.

A subterm convergent rewrite system remains subterm convergent by the addition of a free function symbol *tuple*. Therefore, to compute a finite strongly complete set of variants of a term t it is sufficient to compute a finite complete set of variants of the term $\mathit{tuple}(t, x_1, \ldots x_n)$, where $\mathsf{vars}(t) = \{x_1, \ldots, x_n\}$.

6 Conclusion and further work

We have shown that subterm convergent rewrite systems have the strong finite variant property and we have implemented our algorithm in Section 5 in the prototype tool SubVariant (available at `http://www.lsv.ens-cachan.fr/~ciobaca/subvariant`).

We are currently using this result to obtain a decision procedure for verifying equivalences between cryptographic processes. Another possible direction for future work is to find algorithms for computing strongly complete set of variants modulo associative-commutative function symbols. An extended version of this paper, including the proofs missing due to space constraints is available as a research report [10].

7 Acknowledgements

I would like to thank Steve Kremer, Stéphanie Delaune and the anonymous reviewers for interesting comments regarding this work.

References

1. H. Comon-Lundh and S. Delaune, "The finite variant property: How to get rid of some algebraic properties," in *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, ser. Lecture Notes in Computer Science, J. Giesl, Ed., vol. 3467. Nara, Japan: Springer, Apr. 2005, pp. 294–307. [Online]. Available: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/rta05-CD.pdf
2. S. Escobar, J. Meseguer, and R. Sasse, "Variant narrowing and equational unification," *Electron. Notes Theor. Comput. Sci.*, vol. 238, pp. 103–119, June 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1556507.1556661
3. M. Abadi and V. Cortier, "Deciding knowledge in security protocols under equational theories," in *ICALP*, 2004, pp. 46–58.
4. S. Escobar, J. Meseguer, and R. Sasse, "Effectively checking the finite variant property," in *RTA*, 2008, pp. 79–93.
5. F. Durán, S. Lucas, and J. Meseguer, "Termination modulo combinations of equational theories," in *FroCos*, 2009, pp. 246–262.
6. S. Bursuc and H. Comon-Lundh, "Protocol security and algebraic properties: Decision results for a bounded number of sessions," in *RTA*, 2009, pp. 133–147.
7. S. Bursuc, H. Comon-Lundh, and S. Delaune, "Deducibility constraints, equational theory and electronic money," in *Rewriting, Computation and Proof*, 2007, pp. 196–212.
8. Y. Chevalier and M. Kourjieh, "On the decidability of (ground) reachability problems for cryptographic protocols (extended version)," *CoRR*, vol. abs/0906.1199, 2009.
9. M. Baudet, V. Cortier, and S. Delaune, "YAPA: A generic tool for computing intruder knowledge," in *Proceedings of the 20th International Conference on Rewriting Techniques and Applications (RTA'09)*, ser. Lecture Notes in Computer Science, R. Treinen, Ed., vol. 5595. Brasília, Brazil: Springer, Jun.-Jul. 2009, pp. 148–163. [Online]. Available: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCD-rta09.pdf
10. Ş. Ciobâcă, "Computing finite variants for subterm convergent rewrite systems," Laboratoire Spécification et Vérification, ENS Cachan, France, Research Report LSV-11-06, Apr. 2011, 16 pages. [Online]. Available: http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2011-06.pdf