Computing finite variants for subterm convergent rewrite systems

Ştefan Ciobâcă

April 2011

Research report LSV-11-06

Laboratoire Spécification & Vérification

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

Computing finite variants for subterm convergent rewrite systems

Ştefan Ciobâcă

LSV, ENS Cachan & CNRS

Abstract. We show that subterm convergent rewrite systems enjoy the finite variant property (modulo the empty equational theory). We introduce a variation of the finite variant property, which we call the *strong finite variant property* and we argue that this stronger property is more useful in practice. We also show that the presence at least one free symbol of arity strictly greater than one is a sufficient condition for the two notions to coincide. As an application, we prove that equational unification problems always have a finite complete set of unifiers modulo equational theories with the strong finite variant property.

1 Introduction

Given a term (e.g. $t = \mathbf{dec}(x, y)$) and a convergent rewrite 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. otherwise ($\sigma(x) \neq_{\mathcal{R}} \mathbf{enc}(s, \sigma(y))$ for any term s), in which case $t\sigma\downarrow = t(\sigma\downarrow)$ ($\sigma\downarrow$ 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 [6] and in solving unification and disunification problems [6, 9].

Contributions. In this paper we concentrate on subterm convergent rewrite systems, a class of rewrite systems particularly relevant to security protocol analysis [1]. We show that these rewrite systems have the finite variant property and a slightly stronger property which we call the *strong finite variant property*. We also show that unification problems modulo rewrite systems having the strong finite variant property have a finite complete set of unifiers. The above proofs are constructive and we implement the algorithms for computing a complete finite set of variants of a term and a complete finite set of unifiers for an unification problem in the tool SubVariant.

Related work. The finite variant property was first introduced in [6]. In [8], sufficient conditions and necessary conditions for the finite variant property are introduced. Variant narrowing [9] 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 [7]. Several techniques [3–5] for verifying security protocols make use of the finite variant property as a sub-step in the algorithm.

2 Preliminaries

2.1 Term algebra

We consider the *signature* $\mathcal{F}$ to be a finite set of *function symbols*. We associate to each function symbol $f \in \mathcal{F}$ a natural number $\mathsf{ar}(f)$ which we call the *arity* of f. We consider a countably infinite set $\mathcal{X}$ of *variables* disjoint from $\mathcal{F}$.

If $\mathcal{X}_0 \subseteq \mathcal{X}$, we denote by $\mathcal{T}(\mathcal{X}_0)$ the set of *terms* built inductively from $\mathcal{X}_0$ by applying function symbols from $\mathcal{F}$ (while respecting arities). If $t \in \mathcal{T}(\mathcal{X})$, we denote by $\mathsf{vars}(t)$ the set of variables appearing in t. The function vars is extended as expected to sets of terms.

2.2 Substitutions

Substitutions are mappings from $\mathcal{X}$ to $\mathcal{T}(\mathcal{X})$ such that $\{x \mid \sigma(x) \neq x\}$, the *support* of σ, is finite. We will denote by $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ the substitution of support $\mathsf{sup}(\sigma) = \{x_1, \ldots, x_n\}$ which associates to x_i the term t_i (for all $1 \leq i \leq n$). The *range* $\mathsf{range}(\sigma)$ of a substitution σ is defined to be the set $\mathsf{range}(\sigma) = \{t_i \mid \exists x_i \in \mathsf{sup}(\sigma) \ s.t. \ x_i\sigma = t_i\}$.

The application of a substitution σ to a term t is the term denoted by $t\sigma$, which is obtained from t by replacing all occurrences of all variables x by the term $\sigma(x)$.

A substitution σ is called a *variable renaming* if it is a bijection on $\mathsf{sup}(\sigma)$ and if $\mathsf{range}(\sigma) \subseteq \mathcal{X}$.

The identity substitution is the substitution denoted id, the only substitution with $\mathsf{sup}(\mathsf{id}) = \emptyset$.

The restriction of a substitution σ to a set X of variables is denoted $\sigma[X]$ and is defined such that $\sigma[X](x) = \sigma(x)$ if $x \in X$ and $\sigma[X](x) = x$ if $x \notin \mathcal{X}$.

2.3 Positions

A *position* $p = i_1 \ldots i_n$ (with $n \geq 0$) is a finite sequence of positive natural numbers $i_1, i_2, \ldots, i_n$. The position defined by the empty sequence will be denoted by ϵ. If $p = i_1 \ldots i_n$ and $q = j_1 \ldots j_m$ are two positions, we will denote by $p \cdot q = i_1 \ldots i_n j_1 \ldots j_m$ the concatenation of the two respective sequences. If $\mathcal{P}$ is a set of positions and q is a position, we will denote by $q \cdot \mathcal{P} = \{q \cdot p \mid p \in \mathcal{P}\}$ the set of positions obtained by prefixing each position in $\mathcal{P}$ with q.

The set of *positions* $\mathsf{pos}(t)$ of a term $t \in \mathcal{T}(\mathcal{X})$ is defined inductively as follows:

$$\begin{array}{ll} \mathsf{pos}(t) = \epsilon & \text{if } t = x \in \mathcal{X} \\ \mathsf{pos}(t) = \epsilon \cup \bigcup_{i \in \{1,\ldots,k\}} i \cdot \mathsf{pos}(t_i) & \text{if } t = f(t_1, \ldots, t_k) \text{ where } \mathsf{ar}(f) = k \end{array}$$

The subterm of a term t at position $p \in \mathsf{pos}(t)$ is denoted by $t|_p$ and is defined inductively as follows:

$$\begin{array}{ll} t|_\epsilon = t & \\ t|_{i \cdot p} = t_i|_p & \text{if } t = f(t_1, \ldots, t_k) \text{ where } \mathsf{ar}(f) = k \end{array}$$

The term t with position $p \in \mathsf{pos}(t)$ instantiated to s is denoted $t[s]_p$ and is defined recursively as follows:

$$\begin{array}{ll} t[s]_\epsilon = s & \\ t[s]_{i \cdot p} = f(t_1, \ldots, t_{i-1}, t_i[s]_p, t_{i+1} \ldots t_k) & \text{if } t = f(t_1, \ldots, t_k) \text{ where } \mathsf{ar}(f) = k \end{array}$$

2.4 Subterms

If $t \in \mathcal{T}(\mathcal{X})$, we will denote by $\mathsf{st}(t)$ the set of *subterms* of t, defined to be

$$\mathsf{st}(t) = \{s \mid \exists p \in \mathsf{pos}(t) : s = t|_p\}.$$

If $s \in \mathsf{st}(t)$, we write $s \sqsubseteq t$.

2.5 Syntactic unification

We say that two terms s and t are *unifiable* if there exists a substitution σ of support $\mathsf{sup}(\sigma) = \mathsf{vars}(\{s, t\})$ such that $s\sigma = t\sigma$. Then σ is called a *unifier* of s and t. If σ and τ are unifiers of s and t, σ is called more general than τ if there exists a substitution ω such that $\tau = \sigma \circ \omega$.

It is well known that for every pair of unifiable terms s and t there exists a *most general unifier* σ, i.e. a unifier which is more general than any other unifier of s and t. We will denote by $\mathsf{mgu}(s, t)$ a function which associates to each pair of unifiable terms s and t such a most general unifier.

2.6 Rewriting

A finite set of pairs of terms $\mathcal{R} = \{(l_1, r_1), \ldots, (l_n, r_n)\}$ is called a term rewriting system. We will write $(l, r) \bar{\in} \mathcal{R}$ if there exists $(l', r') \in \mathcal{R}$ and a variable renaming σ with $\mathsf{sup}(\sigma) = \mathsf{vars}(\{l', r'\})$ such that $l = l'\sigma$ and $r = r'\sigma$. This means that (l, r) is in $\mathcal{R}$ up to variable renaming.

We say that t rewrites in one step to s using $\mathcal{R}$ (and we write $t \to_{\mathcal{R}} s$) if there exists a position $p \in \mathsf{pos}(t)$, a pair $(l, r) \bar{\in} \mathcal{R}$ with $\mathsf{vars}(\{l, r\}) \cap \mathsf{vars}(t) = \emptyset$ and a substitution σ such that $t|_p = l\sigma$ and $s = t[r\sigma]_p$.

By $\rightarrow_{\mathcal{R}}^*$ we will denote the transitive and reflexive closure of $\rightarrow_{\mathcal{R}}$. We will say that $\mathcal{R}$ is terminating if there exists no infinite sequence of terms $t_1, \ldots, t_i, \ldots$ such that $t_1 \rightarrow_{\mathcal{R}} \ldots \rightarrow_{\mathcal{R}} t_i \rightarrow_{\mathcal{R}} \ldots$. We will say that $\mathcal{R}$ is confluent if for any terms $t, u, v \in \mathcal{T}(\mathcal{X})$ such that $t \rightarrow_{\mathcal{R}}^* u$ and $t \rightarrow_{\mathcal{R}}^* v$ there exists a term $s \in \mathcal{T}(\mathcal{X})$ such that $u \rightarrow_{\mathcal{R}}^* s$ and $v \rightarrow_{\mathcal{R}}^* s$. A term rewriting system $\mathcal{R}$ is called convergent if it is confluent and terminating.

We say that a term $t \in \mathcal{T}(\mathcal{X})$ is in normal form with respect to $\mathcal{R}$ and we write $t \not\rightarrow_{\mathcal{R}}$ if and only if there exists no $u \in \mathcal{T}(\mathcal{X})$ such that $t \rightarrow_{\mathcal{R}} u$. We say that s is a normal form of t if s is a normal form ($s \not\rightarrow_{\mathcal{R}}$) and if $t \rightarrow_{\mathcal{R}}^* s$. It is well known that if $\mathcal{R}$ is convergent, each term t has an unique normal form which we will denote by $t{\downarrow_{\mathcal{R}}}$. If $\mathcal{R}$ is clear from context, we will also write $t{\downarrow}$ instead of $t{\downarrow_{\mathcal{R}}}$.

We say that a term t is in normal form w.r.t. $\mathcal{R}$ if $t = t{\downarrow_{\mathcal{R}}}$. If $\mathcal{R}$ is clear from context we may simply say that t is in normal form. A substitution σ is in normal form (w.r.t $\mathcal{R}$) if $x\sigma = (x\sigma){\downarrow_{\mathcal{R}}}$ for all variables $x \in \mathsf{sup}(\sigma)$. Again, we may omit $\mathcal{R}$ if it is clear from the context.

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 say that two substitutions σ_1 and σ_2 are *equal modulo* $\mathcal{R}$, and we write $\sigma_1 =_{\mathcal{R}} \sigma_2$ if $\sigma_1(x) =_{\mathcal{R}} \sigma_2(x)$ for all variables x.

In the following sections, we will be 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 $\mathcal{R}$ is convergent and for all $(l, r) \in \mathcal{R}$ we have that:*

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

The first type of rewrite rule justifies the name of this class of rewrite systems and was originally the only type of rule allowed [1] in subterm convergent rewrite systems. The second type of rule represents a small but useful extension introduced in [2]. In this paper, we consider the extended version of subterm convergent rewrite systems as defined above.

3 The finite variant property

We are interested in the following problem:

> Given a term t, compute a finite set of substitutions $\sigma_1, \ldots, \sigma_n$ such that 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.

The substitutions $\sigma_1, \ldots, \sigma_n$ are then called a *complete set of variants* of t.

Example 1. Let $t = \mathbf{dec}(x, y)$ and $\mathcal{R} = \{\mathbf{dec}(\mathbf{enc}(x, y), y) \rightarrow x\}$. 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 sequence of 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*. We show that subterm convergent term rewriting systems have the finite variant property by giving an algorithm that computes such a set for any term t and for a subterm convergent rewrite system $\mathcal{R}$.

4 Algorithm for a complete set of finite variants

In the following we denote by $p\uparrow$ the set of all positions that are descendants of p (including p itself):

$$p\uparrow = \{q \mid q = p \cdot p' \text{ for some } p'\}$$

The algorithm we present for computing a 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.

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

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 will show that:

Theorem 1 (Correctness).
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.

In order to prove the above theorem, we need a stronger invariant which motivates the following technical definition:

Definition 2 (Satisfies).
We say that a tuple (T, ω, τ) (where T is a term and ω and τ are substitutions) satisfies a configuration $(t, \mathcal{P}, \sigma)$ and we write $(T, \omega, \tau) \models (t, P, \sigma)$ if the following conditions hold:

1. $T\omega = (T\sigma)\tau$
2. $T\sigma \to_R^* t$
3. $\forall p \in \mathsf{pos}(t) \setminus \mathcal{P}$ *we have that* $t\tau|_p = (t\tau|_p)\downarrow$
4. $\mathcal{P} \subseteq \mathsf{pos}(t)$

The proof of Theorem 1 easily follows from:

Lemma 1 (Completeness).
If $(T, \omega, \tau) \models (t, \mathcal{P}, \sigma)$, ω is in normal form and $t\tau \neq (t\tau)\downarrow$ there exist p, t', σ', τ' such that

$$(t, \mathcal{P}, \sigma) \hookrightarrow (t', \mathcal{P} \setminus p\uparrow, \sigma')$$

and

$$(T, \omega, \tau') \models (t', \mathcal{P} \setminus p\uparrow, \sigma').$$

Proof. From $(T, \omega, \tau) \models (t, \mathcal{P}, \sigma)$ we have by Definition 2 that:

A. $T\omega = (T\sigma)\tau$
B. $T\sigma \to_{\mathcal{R}}^* t$
C. $\forall p \in \mathsf{pos}(t) \setminus \mathcal{P}$ we have $t\tau|_p = (t\tau|_p)\downarrow$
D. $\mathcal{P} \subseteq \mathsf{pos}(t)$

As $t\tau \neq (t\tau)\downarrow$ we have that there exists $p \in \mathsf{pos}(t\tau)$ such that $t\tau|_p = l\omega'$ for some rewrite rule $l \to r \in \mathcal{R}$ and some substitution ω'. Let p be maximal (w.r.t to its length) with this property. As $\omega = \sigma \circ \tau$ and because ω is in normal form, it follows that τ is in normal form. By (C) and because τ is in normal form, we have that $p \in \mathcal{P}$.

From (D), we have that $t\tau|_p = (t|_p)\tau$. But by the previous observation, we already know that $t\tau|_p = l\omega'$ and therefore $(t|_p)\tau = l\omega'$. This means that $t|_p$ and l are unifiable (as they have the common instance $(t|_p)\tau = l\omega'$). Let $\theta = \mathsf{mgu}(t|_p, l)$. As τ and ω' are instances of the most general unifier θ, we have that there exist τ' and ω'' such that $\tau = \theta \circ \tau'$ and $\omega' = \theta \circ \omega''$.

With our choice of p, $l \to r$ and θ, it follows that a narrowing step can be performed and we obtain that:

$$(t, \mathcal{P}, \sigma) \hookrightarrow (t', \mathcal{P} \setminus p\uparrow, \sigma')$$

where $t' = t\theta[r\theta]_p$ and $\sigma' = \sigma \circ \theta$.

It remains to show that $(T, \omega, \tau') \models (t', \mathcal{P} \setminus p\uparrow, \sigma'))$. It is sufficient to establish that each condition in Definition 2 holds:

1. Firstly we show that $T\omega = (T\sigma')\tau'$. Indeed, by the definition of σ', we have that $(T\sigma')\tau' = (T(\sigma \circ \theta))\tau' = T\sigma\theta\tau' = (T\sigma)(\theta \circ \tau')$ which is equal to $T\sigma\tau$ by the choice of τ' (τ' was chosen such that $\tau = \theta \circ \tau'$). But because of (A) we know that $T\sigma\tau = T\omega$ and therefore we conclude that $T\omega = (T\sigma')\tau'$.
2. Secondly we show that $T\sigma' \to_R^* t'$. We know by (B) that $T\sigma \to_R^* t$. Therefore $T\sigma\theta \to_R^* t\theta$. But $t\theta = t\theta[l\theta]_p$. Therefore $T\sigma\theta \to_R^* t\theta[l\theta]_p$. But $t\theta[l\theta]_p \to_{\mathcal{R}} t\theta[r\theta]_p$ and therefore $T\sigma\theta \to_R^* t\theta[r\theta]_p$, which is exactly our conclusion, since $T\sigma\theta = T\sigma'$ and $t' = t\theta[r\theta]_p$.
3. Next we show that for all $q \in \mathsf{pos}(t') \setminus (\mathcal{P} \setminus p\uparrow)$ we have that $(t'\tau')|_q = (t'\tau')_q\downarrow$. By the definition of t', we have to show that $(t\theta\tau'[r\theta\tau']_p)|_q$ is in normal form for all $q \in \mathsf{pos}(t\theta[r\theta]_p)) \setminus (\mathcal{P} \setminus p\uparrow)$. But $(t\theta\tau'[r\theta\tau']_p)|_q = (t\tau[r\tau]_p)|_q$. We continue by discriminating on the position q:
   (a) if $q \in pos(t) \setminus \{p\}$, we are done by hypothesis (C).
   (b) If $q = p$, we have that $(t\tau[r\tau]_p)|_q = r\tau$ and we show that $r\tau$ is in normal form. We distinguish between two cases:
      i. either $l \to r$ is an extended rule, in which case r and therefore $r\tau$ is ground and in normal form by the definition of subterm convergent rewrite systems.
      ii. or $l \to r$ is a subterm rule, in which case $r\tau$ is a subterm of $l\tau$. Furthermore, as the rewrite system is terminating, $r\tau$ must be a strict subterm of $l\tau$. But by the choice of p and (C), all strict subterms of $t\tau|_p = l\tau$ are in normal form. As $r\tau$ is such a subterm, it is also in normal form.
   (c) if $q \in \mathsf{pos}(t') \setminus (\mathsf{pos}(t) \cup \{p\})$, $t'\tau'|_q$ must be a subterm of $\tau(x)$ for some $x \in \mathsf{sup}(\tau)$. But τ is such that $\omega = \sigma \circ \tau$ and as ω is in normal form, τ must also be in normal form. Therefore $t'\tau'|_q$ is in normal form.

4. Finally we show that $\mathcal{P} \setminus p\uparrow \subseteq \mathsf{pos}(t')$. We have that $\mathcal{P} \setminus p\uparrow \subseteq \mathsf{pos}(t[x]_p)$ (where x is an arbitrary variable) and therefore $\mathcal{P} \setminus p\uparrow \subseteq \mathsf{pos}(t\theta[r\theta]_p)$. But this means $\mathcal{P} \setminus p\uparrow \subseteq \mathsf{pos}(t')$.

Using Lemma 1, we can easily prove Theorem 1:

Proof (of Theorem 1).

Firstly we show that Σ is finite. This is easy to see, since the tree obtained by performing narrowing steps is finitely branching (a narrowing step only depends on the choice of a position and of a rewrite rule) and each branch is finite since the size of the set $\mathcal{P}$ of positions in a configuration strictly decreases with each narrowing step.

Now we show that Σ is a complete set of variants. Let ω be a substitution in normal form. We show that there exists $\sigma \in \Sigma$ such that

$$(t\omega)\downarrow = (t\sigma)\downarrow\tau$$

for some substitution τ.

We have that $(t, \omega, \omega) \models (t, \mathsf{pos}_{init}(t), \mathsf{id})$. By iterating Lemma 1, we obtain that there exist $\tau_i, \mathcal{P}_i, \sigma_i$ $(1 \leq i)$ such that

$$(t, \omega, \tau_i) \models (t_i, \mathcal{P}_i, \sigma_i) \qquad 1 \leq i$$

and such that $|\mathcal{P}_{i+1}| < |\mathcal{P}_i|$ $(1 \leq i)$. As the size of $\mathcal{P}_i$ is strictly decreasing, this sequence must be finite. Let $(t_n, \mathcal{P}_n, \sigma_n)$ be the last configuration in this sequence.

We have that $t_n\tau_n$ is in normal form (otherwise we can apply Lemma 1 and n is not the last index). We show that

$$(t\omega)\downarrow = (t\sigma_n)\downarrow\tau_n.$$

We have that $t\sigma_n \to_R^* t_n$ by Definition 2. Furthermore, $t_n\tau_n$ and therefore t_n is in normal form. Therefore, $(t\sigma_n)\downarrow = t_n$. It remains to prove that

$$(t\omega)\downarrow = t_n\tau_n.$$

We have that $t\omega = (t\sigma_n)\tau_n$ by Definition 2. But $t\sigma_n \to_R^* t_n$ by Definition 2 and therefore $t\omega \to_R^* t_n\tau_n$. But $t_n\tau_n$ is in normal form and therefore the normal form of $t\omega$ is $t_n\tau_n$.

We have shown that

$$(t\omega)\downarrow = (t\sigma_n)\downarrow\tau_n$$

for some $\sigma_n \in \Sigma$ and therefore the proof is complete. □

5 The strong finite variant property

We are interested in the following problem:

Given a term t with $\mathsf{vars}(t) = X$, compute a finite set of substitutions $\sigma_1, \ldots, \sigma_n$ such that 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 $(t\omega)\downarrow = (t\sigma_i)\downarrow\tau$.

The substitutions $\sigma_1, \ldots, \sigma_n$ are then called a *strongly complete set of variants* of t. As with complete sets of variants, a finite such set does not exist in general.

The following example shows that the notion of *complete set of variants* and the notion of *strongly complete set of variants* do not coincide and illustrates the subtlety of the difference.

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 following set S is a complete finite set of variants of t:

$$S = \{\sigma_1 = \mathsf{id}, \sigma_2 = \{x \mapsto f(z)\}\}.$$

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.

5.1 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 see 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.

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

Proof. Indeed, let σ be an arbitrary substitution and t be an arbitrary term.

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

1. If $t{\downarrow} = g^n(f^m(c_k))$, then $(t\sigma){\downarrow} = ((t{\downarrow})\sigma){\downarrow} = t{\downarrow} = (t\mathsf{id}){\downarrow}\mathsf{id}$.
2. If $t{\downarrow} = g^n(f^m(x))$, we consider $s = \sigma(x)$. We have that $s{\downarrow} = g^p(f^q(y))$ or $s{\downarrow} = g^p(f^q(c_r))$ for some variable y and some integers p, q, r.
   (a) if $s{\downarrow} = g^p(f^q(y))$ then $t\sigma{\downarrow} = g^n(f^{m+q}(y)) = (t\mathsf{id}){\downarrow}\{x \mapsto f^q(y)\}$
   (b) if $s{\downarrow} = g^p(f^q(c_r))$ then we have $(t\sigma){\downarrow} = g^n(f^{m+q}(c_r)) = (t\mathsf{id}){\downarrow}\{x \mapsto f^q(c_r)\}$.

We have shown that the identity substitution forms by itself a complete set of variants of any term t over $\mathcal{F}$.

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

Example 4. 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.

5.2 In the presence of free symbols

We have shown in the previous section that in general the strong finite variant property is a strictly stronger property than the finite variant property.

However, if the signature contains at least a free symbol of arity strictly greater than 1, we show that the two notions coincide.

Let $f \in \mathcal{F}$ be the free symbol of arity $k \geq 2$. In the following we will use the notation

$$\mathit{tuple}(t_1, \ldots, t_n) = f(t_1, t, \ldots, t, f(t_2, t, \ldots, t, f(\cdots f(t_n, t, \ldots, t))))$$

where t is an arbitrary ground term in normal form (for example a constant).

Example 5. If $k = 3$, then $\mathit{tuple}(a, b, c) = f(a, t, f(b, t, f(c, t, t)))$.

Because f is a free symbol, we have that

$$\mathit{tuple}(t_1, \ldots, t_n){\downarrow} = \mathit{tuple}(t_1{\downarrow}, \ldots, t_n{\downarrow}) \tag{1}$$

and that

$$\mathit{tuple}(t_1, \ldots, t_n) =_{\mathcal{R}} \mathit{tuple}(s_1, \ldots, s_n) \implies t_1 =_{\mathcal{R}} s_1 \wedge \ldots \wedge t_n =_{\mathcal{R}} s_n. \tag{2}$$

The following theorem states that in the presence of at least a free symbol f of arity $k \geq 2$ the two notions coincide.

Theorem 2. *Let t be a term with $\mathsf{vars}(t) = \{x_1, \ldots, x_n\}$ and let S be a complete set of variants of $\mathit{tuple}(t, x_1, \ldots, x_n)$. Then S is a strongly complete set of variants of t.*

Proof. Let $X = \mathsf{vars}(t)$ and let ω be an arbitrary substitution. We need to show that there exists $\sigma \in S$ such that $\omega[X]{\downarrow} = (\sigma\tau)[X]$ for some substitution τ and that $(t\omega){\downarrow} = (t\sigma){\downarrow}\tau$.

Because S is a set of variants of $\mathit{tuple}(t, x_1, \ldots, x_n)$, we have that there exists $\sigma \in S$ such that

$$(\mathit{tuple}(t, x_1, \ldots, x_n)\omega){\downarrow} = \mathit{tuple}(t, x_1, \ldots, x_n)\sigma){\downarrow}\tau \tag{3}$$

for some substitution τ.

But as a consequence of Equation 1 we have that

$$(\mathit{tuple}(t, x_1, \ldots, x_n)\omega){\downarrow} = \mathit{tuple}(t\omega{\downarrow}, x_1\omega{\downarrow}, \ldots, x_n\omega{\downarrow})$$

and

$$(\mathit{tuple}(t, x_1, \ldots, x_n)\sigma){\downarrow}\tau = \mathit{tuple}((t\sigma){\downarrow}\tau, (x_1\sigma){\downarrow}\tau, \ldots, (x_n\sigma){\downarrow}\tau)$$

and, by making the above two replacements in Equation 3, we obtain

$$\mathit{tuple}(t\omega{\downarrow}, x_1\omega{\downarrow}, \ldots, x_n\omega{\downarrow}) = \mathit{tuple}((t\sigma){\downarrow}\tau, (x_1\sigma){\downarrow}\tau, \ldots, (x_n\sigma){\downarrow}\tau). \tag{4}$$

By applying Equation 2 to Equation 4 we immediately obtain that $(t\omega){\downarrow} = (t\sigma){\downarrow}\tau$ and that $\omega[X]{\downarrow} = (\sigma{\downarrow}\tau)[X]$. □

Corollary 1. *Subterm convergent rewrite systems have the strong finite variant property.*

Proof. By adding a free symbol to the signature the rewrite system remains subterm convergent. Therefore, by Theorem 2 and Theorem 1, we immediately conclude. □

6 Equational unification

Given two terms s and t, the problem of unifying s and t modulo $\mathcal{R}$ is denoted

$$s \doteq_{\mathcal{R}} t.$$

A *unifier modulo* $\mathcal{R}$ of s and t is any substitution σ such that

$$s\sigma =_{\mathcal{R}} t\sigma.$$

When σ is a unifier modulo $\mathcal{R}$ of s and t, we also say that σ is a solution of $s \doteq_{\mathcal{R}} t$.

Example 6 (equational unification).

Consider the unification problem

$$\mathbf{dec}(x, k) \doteq_{\mathcal{R}} y$$

where $\mathcal{R} = \{\mathbf{dec}(\mathbf{enc}(x, y), y) \to x\}$ and where k is a constant symbol. The substitution

$$\sigma = \{x \mapsto \mathbf{enc}(y, k)\}$$

is then a solution modulo $\mathcal{R}$ of the above equation.

The goal of equational unification is, given a unification problem

$$s \doteq_{\mathcal{R}} t,$$

to characterize all of its solutions. In the case of *syntactic unification* (i.e. where $\mathcal{R}$ is empty), this is rather easy, since any two unifiable terms have a most general unifier, in the sense that any other solution is obtained by instantiating the most general unifier.

In the case of equational unification, the notion of most general unifier is no longer sufficient since two terms unifiable modulo $\mathcal{R}$ need not have a most general unifier modulo $\mathcal{R}$:

Example 7 (no most general unifier in the case of equational unification).
Let $\mathcal{R} = \{f(x, y, 0) \to x, f(x, y, 1) \to y\}$ and consider the unification problem

$$f(g(a), g(b), x) = g(y).$$

We have that $\sigma_1 = \{x \mapsto 0, y \mapsto a\}$ is a solution modulo $\mathcal{R}$ of this equation. However, $\sigma_2 = \{x \mapsto 1, y \mapsto b\}$ is also a solution. It can bee seen that σ_1 and σ_2 are not instances of any more general solution of the equation. This shows that the notion of most general unifier does not make sense in the context of equational unification.

Therefore, the notion of most general unifier is replaced in the case of equational unification by the notion of a *complete set of unifiers*.

Definition 3 (complete set of unifiers).
Let

$$s \doteq_{\mathcal{R}} t$$

be a unification problem. A set $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ of substitutions is called a complete set of unifiers of $s \doteq_{\mathcal{R}} t$ if:

1. *any substitution σ_i $(1 \leq i \leq n)$ is a solution of $s \doteq_{\mathcal{R}} t$, i.e. $s\sigma_i =_{\mathcal{R}} t\sigma_i$.*
2. *any solution σ of $s \doteq_{\mathcal{R}} t$ is an instance modulo $\mathcal{R}$ of σ_i for some $1 \leq i \leq n$ (i.e. $\exists \tau : \sigma[X] =_{\mathcal{R}} (\sigma_i \tau)[X]$).*

It is well known that there exist rewriting systems for which finite complete set of unifiers do not exist.

6.1 Algorithm

We show next that if $\mathcal{R}$ has the strong finite variant property, a unification problem modulo $\mathcal{R}$ always has a finite complete set of unifiers and we give an algorithm to obtain such a set.

We consider a free function symbol f which does not appear in Σ (and therefore not in $\mathcal{R}$). For any equation $s \doteq_{\mathcal{R}} t$, we let $\mathcal{V}(s, t)$ be a strongly complete set of variants of $f(s, t)$.

Then the set

$$\mathcal{U}(s,t) = \{\sigma\omega \mid \sigma \in \mathcal{V}(s,t), \omega = \mathsf{mgu}(s\sigma\downarrow, t\sigma\downarrow)\}$$

is a complete set of unifiers of s and t modulo $\mathcal{R}$.

The unification algorithm consists then of computing the complete set of unifiers $\mathcal{U}(s,t)$ starting from the strongly complete set of variants $\mathcal{V}(s,t)$ of $f(s,t)$ by applying syntactic unification as described above.

Theorem 3 (soundness and completeness of the algorithm).
$\mathcal{U}(s,t)$ is a complete set of unifiers of s and t.

Proof. We first show the soundness of the algorithm: for any $\psi \in \mathcal{U}(s,t)$, we have that ψ is a solution of $s \doteq_{\mathcal{R}} t$.

As $\psi \in \mathcal{U}(s,t)$, we have that there exist $\sigma \in \mathcal{V}(s,t)$ and $\omega = \mathsf{mgu}(s\sigma\downarrow, t\sigma\downarrow)$ such that $\psi = \sigma\omega$. Therefore $s\sigma\downarrow\omega = t\sigma\downarrow\omega$, which immediately implies $s\sigma\omega =_{\mathcal{R}} t\sigma\omega$, i.e. $s\psi = t\psi$.

Let $X = \mathsf{vars}(s,t)$. Next we show the completeness of the algorithm: for any substitution τ such that $s\tau =_{\mathcal{R}} t\tau$ we have that there exists $\psi \in \mathcal{U}(s,t)$ such that $\tau[X] =_{\mathcal{R}} (\psi\phi)[X]$ for some substitution ϕ.

By the strong completeness of the set of variants $\mathcal{V}(s,t)$, it follows that there exists $\sigma \in \mathcal{V}(s,t)$ such that $f(s,t)\tau\downarrow = f(s,t)\sigma\downarrow\tau'$ and that $\tau[X] = (\sigma\downarrow\tau')[X]$ for some substitution τ'. Furthermore $f(s,t)\sigma\downarrow\tau' = f(s\sigma\downarrow\tau', t\sigma\downarrow\tau')$ and $f(s,t)\tau\downarrow = f(s\tau\downarrow, t\tau\downarrow)$ as f is a free symbol.

By transitivity, we obtain that $f(s\tau\downarrow, t\tau\downarrow) = f(s\sigma\downarrow\tau', t\sigma\downarrow\tau')$ and therefore $s\tau\downarrow = s\sigma\downarrow\tau'$ and $t\tau\downarrow = t\sigma\downarrow\tau'$.

As $s\tau =_{\mathcal{R}} t\tau$, we have that $s\tau\downarrow = t\tau\downarrow$ and by transitivity we obtain $s\sigma\downarrow\tau' = t\sigma\downarrow\tau'$. As τ' is a unifier of $s\sigma\downarrow$ and $t\sigma\downarrow$, we have that there exists $\omega = \mathsf{mgu}(s\sigma\downarrow, t\sigma\downarrow)$ and that τ' is an instance of ω: $\exists\tau''$ such that $\tau' = \omega\tau''$.

Let $\psi = \sigma\omega$ and $\phi = \tau''$. By the definition of $\mathcal{U}(s,t)$, we have that $\psi \in \mathcal{U}(s,t)$. All that is left to show is that $\tau[X] =_{\mathcal{R}} (\psi\phi)[X]$, or equivalently, that $\tau[X] =_{\mathcal{R}} (\sigma\omega\tau'')[X]$ (we replaced ϕ with its definition). But $\omega\tau'' = \tau'$ and therefore we only need to show that $\tau[X] =_{\mathcal{R}} (\sigma\tau')[X]$. But we already know that $\tau[X] = (\sigma\downarrow\tau')[X]$ (from the choice of σ and τ') and therefore we immediately get the equality modulo $\mathcal{R}$: $\tau[X] =_{\mathcal{R}} (\sigma\tau')[X]$.

7 Conclusion and further work

We have shown that subterm convergent rewrite systems have the strong finite variant property. We have implemented our algorithm in Section 4 in the prototype tool SubVariant (available at `http://www.lsv.ens-cachan.fr/~ciobaca/subvariant`). It receives as input a subterm convergent rewrite system $\mathcal{R}$ and a term t and it outputs a complete finite set Σ of variants of t. The tool is also able to compute a complete set of unifiers of an equation modulo $\mathcal{R}$.

As future work, we intend to use this result to obtain a decision procedure for verifying equivalences between cryptographic processes.

Another direction for future work is to investigate algorithms for computing finite variants modulo equational theories containing associative-commutative function symbols.

8 Acknowledgements

I would like to thank Steve Kremer for carefully reading a draft of this research report and offering many suggestions for improvements and Stéphanie Delaune for an interesting discussion regarding the definition of the finite variant property which helped shape this research report.

References

1. M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. In *ICALP*, pages 46–58, 2004.
2. M. Baudet, V. Cortier, and S. Delaune. YAPA: A generic tool for computing intruder knowledge. In R. Treinen, editor, *Proceedings of the 20th International Conference on Rewriting Techniques and Applications (RTA'09)*, volume 5595 of *Lecture Notes in Computer Science*, pages 148–163, Brasília, Brazil, June-July 2009. Springer.
3. S. Bursuc and H. Comon-Lundh. Protocol security and algebraic properties: Decision results for a bounded number of sessions. In *RTA*, pages 133–147, 2009.
4. S. Bursuc, H. Comon-Lundh, and S. Delaune. Deducibility constraints, equational theory and electronic money. In *Rewriting, Computation and Proof*, pages 196–212, 2007.
5. Y. Chevalier and M. Kourjieh. On the decidability of (ground) reachability problems for cryptographic protocols (extended version). *CoRR*, abs/0906.1199, 2009.
6. H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In J. Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307, Nara, Japan, Apr. 2005. Springer.
7. F. Durán, S. Lucas, and J. Meseguer. Termination modulo combinations of equational theories. In *FroCos*, pages 246–262, 2009.
8. S. Escobar, J. Meseguer, and R. Sasse. Effectively checking the finite variant property. In *RTA*, pages 79–93, 2008.
9. S. Escobar, J. Meseguer, and R. Sasse. Variant narrowing and equational unification. *Electron. Notes Theor. Comput. Sci.*, 238:103–119, June 2009.