On Problems Dual to Unification

Zümrüt Akçam[1], Daniel S. Hono II[1], and Paliath Narendran[1]

University at Albany, SUNY
Albany, NY, US
{zakcam, dhono, pnarendran}@albany.edu

Abstract

In this paper, we investigate a problem dual to the unification problem, namely the *Common Term (CT)* problem for string rewriting systems. Our main motivation is computing *fixed points* in systems, such as *loop invariants* in programming languages. We show that the fixed point problem is reducible to the common term problem. We also prove that the common term problem is undecidable for the class of *dwindling* string rewriting systems.

1 Introduction

In every major research field, there are variables or other parameters that change over time. These variables are modified — increased or decreased — as a result of a change in the environment. Computing *invariants,* or expressions whose values do not change under a transformation, is very important in many areas such as Physics, e.g., invariance under the *Lorentz* transformation.

In Computer Science, the issue of obtaining invariants arises in *axiomatic semantics* or *Floyd-Hoare semantics*, in the context of formally proving a loop to be correct. A *loop invariant* is a condition, over the program variables, that holds before and after each iteration. Our research is partly motivated by the related question of finding expressions, called *fixed points*, whose values will be the same before and after each iteration, i.e., will remain unchanged through each iteration.

We formulate this problem in terms of properties of substitutions *modulo* a term rewriting system. One straightforward formulation is as follows:

Input: A substitution θ and an equational theory E.

Question: Does there exist a non-ground term $t \in T(\Sigma, \mathscr{D}om(\theta))$ such that $\theta(t) \approx_E t$?

The decision problem above is referred to as the *fixed point problem* (FP).

Example 1: Suppose E is a theory of integers which contains linear arithmetic. Let $\theta = \{x \mapsto x-2, y \mapsto y+1\}$ and we would like to find a term t such that $\theta(t) \approx_E t$. Note that $x+2y$ is such a term, since

$$\theta(x+2y) = (x-2)+2*(y+1) \approx_E x+2y$$

We plan to explore two related formulations, both of which can be viewed as *dual* to the well-known unification problem. Unification deals with solving symbolic equations: a typical input would be either two terms, say s and t, or an equation $s \approx^? t$. The task is to find a substitution such that $\theta(s) \approx \theta(t)$. For example, given two terms $s_1 = f(a,y)$ and $s_2 = f(x,b)$, where f is a binary function symbol, a and b are constants, and x and y are variables, the substitution $\sigma = \{x \mapsto a,\ y \mapsto b\}$ unifies s_1 and s_2, or equivalently, σ is a unifier for the equation $s_1 =^? s_2$.

There are two ways to "dualize" the unification problem, *common term (CT) problem* and *common equation (CE) problem* respectively:

Input: Two ground substitutions θ_1 and θ_2, and an equational theory E. (i.e., $\mathscr{VR}an(\theta_1) = \emptyset$ and $\mathscr{VR}an(\theta_2) = \emptyset$)

Question: Does there exist a non-ground term $t \in T(\Sigma, \mathscr{D}om(\theta_1) \cup \mathscr{D}om(\theta_2))$ such that $\theta_1(t) \approx_E \theta_2(t)$?

Example 2: Consider the two substitutions $\theta_1 = \{x \mapsto p(a), y \mapsto p(b)\}$ and $\theta_2 = \{x \mapsto a, y \mapsto b\}$. If we take the term rewriting system R_1^{lin} in the appendix as our background equational theory E, then there exists a common term $t = x - y$ that satisfies $\theta_1(t) \approx_E \theta_2(t)$.

$$\theta_1(x-y) \approx_E p(a) - p(b) \approx_E a - b \text{ and } \theta_2(x-y) \approx_E a - b$$

We can easily show that the fixed point problem can be reduced to the CT problem.

Lemma 1. *The fixed point problem is reducible to the common term problem.*

Proof. Let θ_2 be the empty substitution. Assume that the fixed point problem has a solution, i.e., there exists a term t such that $\theta(t) \approx_E t$. Then the CT problem for θ and θ_2 has a solution since $\theta_2(t) \approx_E t$ (because $\theta_2(s) = s$ for all s). The "only if" part is trivial, again because $\theta_2(s) = s$ for all s.

Alternatively, suppose that $\mathscr{D}om(\theta)$ consists of n variables, where $n \geq 1$. If we map all the variables in $\mathscr{VR}an(\theta)$ to new constants, this will create a ground substitution $\theta_1 = \{x_1 \mapsto a_1, x_2 \mapsto a_2, \ldots, x_n \mapsto a_n\}$. θ_1 will be the one of the substitutions for the CT problem. The other substitution, θ_2, is the composition of the substitutions θ and θ_1. The substitution θ_1 will replace all of the variables in $\mathscr{VR}an(\theta)$ with the new constants, thus making θ_2 a ground substitution. Now if $\theta(t) \approx_E t$, then $\theta_2(t) = \theta_1(\theta(t)) \approx_E \theta_1(t)$; in other words, t is a solution to the common term problem.

The "only if" part can also be explained in terms of the composition above. Suppose that $\theta_1(s)$ and $\theta_2(s)$ are equivalent, i.e., $\theta_1(s) \approx_E \theta_2(s)$ for some s. Since $\theta_2 = \theta_1 \circ \theta$, the equation can be rewritten as $\theta_1(\theta(s)) \approx_E \theta_1(s)$. Since $a_1, \ldots, a_n$ are new constants and are not included in the signature of the theory, for all t_1 and t_2, $\theta_1(t_1) \approx_E \theta_1(t_2)$ holds if and only if $t_1 \approx_E t_2$ (See [3], Section 4.1, page 60) Thus $\theta_1(\theta(s)) \approx_E \theta_1(s)$ implies that $\theta(s) \approx_E s$, making s a fixed point. □

Input: Two substitutions θ_1 and θ_2 with the *same domain,* and an equational theory E.

Question: Does there exist a non-ground, non-trivial $(t_1 \not\approx_E t_2)$ equation $t_1 \approx^? t_2$, where $t_1, t_2 \in T(\Sigma, \mathscr{D}om(\theta_1))$ such that both θ_1 and θ_2 are E-unifiers of $t_1 \approx^? t_2$?

By trivial equations, we mean equations which are identities in the equational theory E, i.e., an equation $s \approx^?_E t$ is trivial if and only if $s \approx_E t$. We exclude this type of trivial equations in the formulation of this question.

Example 3: Let $E = \{p(s(x)) \approx x,\ s(p(x)) \approx x\}$. Given two substitutions $\theta_1 = \{x_1 \mapsto s(s(a)), x_2 \mapsto s(a)\}$ and $\theta_2 = \{x_1 \mapsto s(a), x_2 \mapsto a\}$, we can see that $\theta_1(t_1) \approx_E \theta_1(t_2)$ and $\theta_2(t_1) \approx_E \theta_2(t_2)$, with the equation

$$p(x_1) \approx_E x_2$$

However, there is no term t on which the substitutions agree, i.e., there aren't any solutions for the common term problem in this example. Thus, CT and CE problems are not equivalent as we observe in the example above.

We investigate these two problems for the string rewriting case. Due to the lack of space, we only focus on the *Common Term (CT) problem* in this paper. For further details, please refer to the technical report [2].

2 Definitions

We start by presenting some definitions for the string rewriting systems that we use in this paper; for more details, refer to the books [3] for term rewriting systems and to [6] for string rewriting systems. Remember that concepts such as *normal form, terminating, confluent,* and *convergent* have the same definitions in the string rewriting systems as they have for the term rewriting systems.

String rewriting systems are equivalent to term rewriting systems where all function symbols are of arity at most 1. Strings are viewed as terms where letters correspond to function symbols applied to a variable in the order they appear in the string. (In other words, concatenation is viewed as composition.) For example, if $g, h \in \Sigma$, then the string gh will be seen as the term $h(g(x))$[1]

A string rewriting system T is said to be: ***monadic*** iff the rhs of each rule in T is either a single symbol or the empty string, e.g., $abc \to b$; ***dwindling*** iff, for every rule $l \to r$ in T, the rhs r is a *proper prefix* of its lhs l, as, for example, in $abc \to ab$, and ***length-reducing*** iff $|l| > |r|$ for all rules $l \to r$ in T, e.g., $abc \to ba$.

3 Common Term (CT) Problem

We explore the common term problem for the class of (general) convergent string rewriting systems and *three* subclasses: length-reducing, dwindling and monadic.

It is known that the CT problem is undecidable for convergent string rewriting systems; in fact, Otto et al. [12] proved that the CT problem is undecidable even for *convergent and length-reducing* string rewriting systems. However, they also proved that the CT problem is decidable in polynomial time for convergent and monadic string rewriting systems (see, e.g., [12], Lemma 3.7).

In this paper, we focus on the decidability of the CT problem for *convergent and dwindling* string rewriting systems. The dwindling convergent systems are especially important because they are widely used in the field of protocol analysis; in particular, digital signatures, one-way hash functions and standard axiomatization of encryption and decryption. This class is also known as subterm convergent theories in the literature [1, 4, 8, 9]. Tools such as TAMARIN prover [10] and YAPA [5] use subterm-convergent theories since these theories have nice properties (e.g., finite basis property [7]) and decidability results [1].

3.1 CT for Dwindling Systems

We define CT (Common Term) as the following decision problem:

Given: A finite, non-empty alphabet Σ, strings $\alpha, \beta \in \Sigma^*$ and a dwindling, convergent string rewriting system S.

Question: Does there exist a string $W \in \Sigma^*$ such that $\alpha W \approx_S \beta W$?

Note that interpreting concatenation the other way, i.e., ab as $a(b(x))$, will make this a *unification* problem.

We show that Generalized Post Correspondence Problem (*GPCP*) reduces to the CT problem, where *GPCP* stands for a slight change to the modified post correspondence problem such that we will provide

[1] It may be more common to view gh as $g(h(x))$ with function application done in the reverse order.

the start and finish dominoes in the problem instance. This slight change does not affect the decidability of the problem in any way, i.e., *GPCP* is also undecidable [11].

Given: A finite set of tuples $\{(x_i,\, y_i)\}_{i=0}^{n+1}$ such that each $x_i, y_i \in \Sigma^+$, i.e. for all i, $|x_i| > 0$, $|y_i| > 0$, and $(x_0, y_0), (x_{n+1}, y_{n+1})$ are start and end dominoes, respectively.

Question: Does there exist a sequence of indices $i_1, \ldots, i_k$ such that

$$x_0\, x_{i_1}\, \ldots\, x_{i_k}\, x_{n+1} = y_0\, y_{i_1}\, \ldots\, y_{i_k}\, y_{n+1}?$$

We work towards showing that the CT problem defined above is undecidable by a many-one reduction from *GPCP*. First, we show how to construct a string-rewriting system that is dwindling and convergent from a given instance of *GPCP*.

Let $\{(x_i,\, y_i)\}_{i=1}^{n}$ the set of "intermediate" dominoes and $(x_0, y_0), (x_{n+1}, y_{n+1})$, the start and end dominoes respectively, be given. Suppose Σ is the alphabet given in the instance of GPCP, and without loss of generality, we may assume $\Sigma = \{a, b\}$. Then set $\hat{\Sigma} := \{a, b\} \cup \{c_0,\, ..\, c_{n+1}\} \cup \{\dot{\varsigma}_1, \dot{\varsigma}_2, B, a_1, a_2, a_3, b_1, b_2, b_3\}$ which will be our alphabet for the instance of CT.

Next we define a set of string homomorphisms used to simplify the discussion of the reduction. Namely, we have the following:

$$\begin{array}{llllll} h_1(a) = & a_1\, a_2\, a_3, & h_2(a) = & a_1\, a_2, & h_3(a) = & a_1 \\ h_1(b) = & b_1\, b_2\, b_3, & h_2(b) = & b_1\, b_2, & h_3(b) = & b_1 \end{array}$$

such that each $h_i : \Sigma \to \hat{\Sigma}^+$ is a homomorphism.

We are now in a position to construct the string rewriting system, S, with the following collections of rules, named as the Class D rules:

$$\begin{array}{llllll} \dot{\varsigma}_1 h_1(a) \to & \dot{\varsigma}_1 h_3(a), & \dot{\varsigma}_2 h_1(a) \to & \dot{\varsigma}_2 h_2(a) \\ \dot{\varsigma}_1 h_1(b) \to & \dot{\varsigma}_1 h_3(b), & \dot{\varsigma}_2 h_1(b) \to & \dot{\varsigma}_2 h_2(b) \end{array}$$

and,

$$\begin{array}{llllll} h_i(a)\, h_1(a) \to & h_i(a)\, h_i(a), & h_i(a)\, h_1(b) \to & h_i(a)\, h_i(b) \\ h_i(b)\, h_1(a) \to & h_i(b)\, h_i(a), & h_i(b)\, h_1(b) \to & h_i(b)\, h_i(b) \end{array}$$

for $i \in \{2, 3\}$.

The erasing rules of our system consist of three classes. Class I, Class II and Class III rules are defined as follows, for each $i = 1, 2, \ldots, n$:

Class I	Class II	Class III
$\dot{\varsigma}_1\, h_3(x_0)\, B\, c_0 \to \lambda$	$h_3(x_i)\, B\, c_i \to \lambda$	$h_3(x_{n+1})\, c_{n+1} \to \lambda$
$\dot{\varsigma}_2\, h_2(y_0)\, c_0 \to \lambda$	$h_2(y_i)\, c_i\, B \to \lambda$	$h_2(y_{n+1})\, c_{n+1}\, B \to \lambda$

Table 1: Classes of Erasing Rules.

Clearly, given an instance of *GPCP*, the above set of rules can effectively be constructed from the instance data. Also, by inspection, we have that our system is confluent (there are no overlaps between left-hand sides of any rules), terminating, and dwindling.

We then set $\alpha = \dot{c}_1$ and $\beta = \dot{c}_2$ to complete the constructed instance of *CT* from *GPCP*.

It remains to show that this instance of *CT* is a "yes" instance if and only if the given instance of *GPCP* is a "yes" instance. In that direction, we prove some results relating to *S*.

Lemma 2. *Suppose* $\dot{c}_1 h_3(w_1) B\gamma \to^! \lambda$ *and* $\dot{c}_2 h_2(w_2)\gamma \to^! \lambda$ *for some* $w_1, w_2 \in \{a,b\}^*$, *then* $\gamma \in \{c_1B,\ c_2B,\ \ldots, c_nB\}^* c_0$.

Proof. Suppose γ is a minimal counter example with respect to length and $\gamma \in IRR(R)$. In order for the terms to be reducible, $\gamma = c_i B\ \gamma'$ (this follows by inspection of *S*). After we replace the γ at the equation in the lemma, we get:

$$\begin{aligned} \dot{c}_1\ h_3(w_1)\ B\ c_i\ B\ \gamma' &\to \dot{c}_1 h_3(w_1)' B\ \gamma' \to^! \lambda \\ \dot{c}_2\ h_2(w_2)\ c_i\ B\ \gamma' &\to \dot{c}_2 h_2(w_2)'\ \gamma' \to^! \lambda \end{aligned}$$

by applying the Class II rules and finally Class I rule to erase the $\dot{c}$ signs. Then, however, γ' is also a counterexample, and $|\gamma'| < |\gamma|$, which is a contradiction. □

We are now in a position to state and prove the main result of this section.

Theorem 1. *The CT problem is undecidable for dwindling convergent string-rewriting systems.*

Proof. We first complete the "only if" direction. Suppose CT has a solution such that $\dot{c}_1 Z \downarrow \dot{c}_2 Z$ where Z is a minimal solution. We show that Z corresponds to a solution for *GPCP*. Let $Z = h_1(Z_1)Z_2$ such that $h_1(Z_1)$ is the longest prefix of Z such that the following relationship holds: $Z = Z'\ Z_2$ and $Z' = h_1(Z_1)$ for some string Z_1.

$h_1(Z_1)$ can be rewritten to $h_3(Z_1)$ and $h_2(Z_1)$ by applying the Class D rules. Thus, we will get

$$\begin{aligned} \dot{c}_1\ h_1(Z_1)\ Z_2 &\to^* \dot{c}_1 h_3(Z_1)\ Z_2 \\ \dot{c}_2\ h_1(Z_1)\ Z_2 &\to^* \dot{c}_2 h_2(Z_1)\ Z_2 \end{aligned}$$

In order for the terms to be reducible simultaneously, Z_2 must be of the form $Z_2 = c_{n+1}\ B\ Z_2'$. After replacing Z_2 with its equivalent, Z_1 can be rewritten with i.e. $Z_1 = Z_1'\ x_{n+1}$ and $Z_1 = Z_1''\ y_{n+1}$. Thus, by applying the Class III rules, these equations will reduce to:

$$\begin{aligned} \dot{c}_1\ h_3(Z_1)\ Z_2 &= \dot{c}_1 h_3(Z_1)\ c_{n+1}\ B\ Z_2' \to \dot{c}_1 h_3(Z_1)'\ B\ Z_2' \\ \dot{c}_2\ h_2(Z_1)\ Z_2 &= \dot{c}_2 h_2(Z_1)\ c_{n+1}\ B\ Z_2' \to \dot{c}_2 h_2(Z_1)''\ Z_2' \end{aligned}$$

We now apply Lemma 2 to conclude that $Z_2' \in \{c_1B,\ c_2B,\ \ldots, c_nB\}^* c_0$.

At this point we have that, $Z_2 = c_{n+1}Bc_{i_1}Bc_{i_2}\cdots Bc_{i_k}Bc_0$ for some $i_1, \ldots, i_k$

Then the sequence of dominoes, $(x_0,y_0),(x_{i_k},y_{i_k}),\ldots,(x_{i_1},y_{i_1}),(x_{n+1},y_{n+1})$ will be a solution to the given instance of *GPCP* with solution string Z_1 since the LHS of the Class I, II, III rules consist of the images of domino strings under h_2 and h_3. More specifically, there is a finite number of B's and c_i's in Z_2, so there must be a decomposition of $h_1(Z_1)$:

$$h_1(Z_1) = h_1(x_0)h_1(x_{i_1})\cdots h_1(x_{i_k})h_1(x_{n+1}) \ \text{ and } \ h_1(Z_1) = h_1(y_0)h_1(y_{i_1})\cdots h_1(y_{i_k})h_1(y_{n+1})$$

Thus, first we have the following reductions with Class D rules and then, finally, by Class I, II, III rules:

$$\dot{\varsigma}_1\ h_1(Z_1)\ Z_2 \to^* \dot{\varsigma}_1\ h_3(Z_1)\ Z_2 \to^* \dot{\varsigma}_1\ h_3(x_0)\ B\ c_0 \to \lambda$$

$$\dot{\varsigma}_2\ h_1(Z_1)\ Z_2 \to^* \dot{\varsigma}_2\ h_2(Z_1)\ Z_2 \to^* \dot{\varsigma}_2\ h_2(y_0)\ c_0 \to \lambda$$

and Z_1 is a solution to the instance of the *GPCP*.

We next prove the "if" direction. Assume that the given instance of *GPCP* has a solution. Let w be the string corresponding to the matching dominoes, and let $(x_0,y_0)(x_{i_1},y_{i_1})\cdots(x_{i_k},y_{i_k})(x_{n+1},y_{n+1})$ be the sequence of tiles that induces the match. Let $Z = c_{n+1}Bc_{i_1}Bc_{i_2}\cdots Bc_{i_k}Bc_0$, then we show that $\dot{\varsigma}_1 h_1(w)Z \downarrow \dot{\varsigma}_2 h_1(w)Z$.

First apply the Class *D* rules to get:

$$\dot{\varsigma}_1 h_1(w)Z \to^* \dot{\varsigma}_1 h_3(w)Z$$

$$\dot{\varsigma}_2 h_1(w)Z \to^* \dot{\varsigma}_2 h_2(w)Z$$

but then we can apply class I, II, III rules to reduce both of the above terms to λ. □

Acknowledgements: We thank the referees for their thoroughness and their comments and suggestions which were very helpful in improving the paper.

References

[1] Martín Abadi and Véronique Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 367(1-2):2–32, 2006.

[2] Zümrüt Akçam, Daniel S. Hono II, and Paliath Narendran. On problems dual to unification. Technical report, University at Albany, SUNY, 2017. Accessible at https://arxiv.org/abs/1706.05607.

[3] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1999.

[4] Mathieu Baudet. Deciding security of protocols against off-line guessing attacks. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, CCS '05, pages 16–25, New York, NY, USA, 2005. ACM.

[5] Mathieu Baudet, Véronique Cortier, and Stéphanie Delaune. *YAPA: A Generic Tool for Computing Intruder Knowledge*, pages 148–163. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[6] Ronald V. Book and Friedrich Otto. *String-rewriting systems*. Springer, 1993.

[7] Yannick Chevalier and Michaël Rusinowitch. Compiling and securing cryptographic protocols. *Information Processing Letters*, 110(3):116–122, 2010.

[8] Ştefan Ciobâcă, Stéphanie Delaune, and Steve Kremer. *Computing Knowledge in Security Protocols under Convergent Equational Theories*, pages 355–370. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[9] V. Cortier and S. Delaune. A method for proving observational equivalence. In *2009 22nd IEEE Computer Security Foundations Symposium*, pages 266–276, July 2009.

[10] Simon Meier, Benedikt Schmidt, Cas Cremers, and David Basin. *The TAMARIN Prover for the Symbolic Analysis of Security Protocols*, pages 696–701. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[11] François Nicolas. (Generalized) Post Correspondence Problem and semi-Thue systems. *CoRR*, abs/0802.0726, 2008.

[12] Friedrich Otto, Paliath Narendran, and Daniel J. Dougherty. Equational unification, word unification, and 2nd-order equational unification. *Theoretical Computer Science*, 198(1-2):1–47, 1998.

A Appendix

Definition 1. *The following term rewriting system R_1^{lin} specifies a fragment of linear arithmetic using* successor *and* predecessor *operators:*

$$\begin{array}{rcl} x-0 & \to & x \\ x-x & \to & 0 \\ s(x)-y & \to & s(x-y) \\ p(x)-y & \to & p(x-y) \\ x-p(y) & \to & s(x-y) \\ x-s(y) & \to & p(x-y) \\ p(s(x)) & \to & x \\ s(p(x)) & \to & x \end{array}$$

This TRS is convergent.