Equational Unification, Word Unification, and 2^{nd}-Order Equational Unification*

Friedrich Otto

Fachbereich Mathematik/Informatik, Universität Kassel
34109 Kassel, Germany
Internet: otto@theory.informatik.uni-kassel.de

Paliath Narendran†

Institute of Programming and Logics, Department of Computer Science
State University of New York, Albany, NY 12222, U.S.A.
Internet: dran@cs.albany.edu

Daniel Dougherty

Mathematics Department, Wesleyan University, Middletown, CT 06459, U.S.A.
Internet: ddougherty@wesleyan.edu

November 15, 1996

Abstract

For finite convergent term-rewriting systems it is shown that the *equational unification problem* is recursively independent of the *equational matching problem*, the *word matching problem*, and the 2^{nd}*-order equational matching problem*. Apart from the latter these results are derived by considering term-rewriting systems on signatures that contain unary function symbols only (that is, string-rewriting systems). Also for this special case 2^{nd}-order equational matching is shown to be reducible to 1^{st}-order equational matching. In addition, we present some new decidability results for *simultaneous equational matching and unification*. Finally, we compare the *word unification problem* to the 2^{nd}*-order equational unification problem*.

1 Introduction

Unification deals with the problem of solving an equation of the form $s = t$ in the free algebra $T(F, X)$ of terms that are generated from the function symbols in F (each of them with a fixed arity) and the variables in X. As is well-known such an equation either has no solution at all, or there exists a single *most general* solution [Rob65], that is, each solution is an instance of this most general one. Also algorithms are known that determine this most general solution in linear time from the given equation [PaWe78] . Unification is an essential tool in logic programming and machine-oriented logic, where it is the basic mechanism underlying the

*Some of the results of this paper have been presented at the 6^{th} International Conference on Rewriting Techniques and Applications (RTA-95) at Kaiserslautern, April 1995.

†The work of this author was partially supported by the NSF grant CCR-9404930.

resolution principle, and it is fundamental in the rewrite-rule based approach to equational reasoning as embodied in the Knuth-Bendix completion procedure [AvMa90].

Often we are interested in reasoning about terms modulo a 1^{st}-order equational theory, and it has proved fruitful in many cases to "build-in" such a theory in the unification process. This leads to the notion of *equational unification,* E-unification for short, the problem of solving an equation in (the initial model for) the given equational theory. Because of its importance for logic programming and equational reasoning, this problem has received a great deal of attention in the literature; see [BaSi93] for a recent survey of the area.

We can generalize the syntactic unification problem in another direction, by considering terms which may contain function variables, eligible for instantiation. For example, if a is an individual constant, f a function symbol, and v a function variable, then the unification problem $f(a) = v(a)$ has the solution in which v is instantiated by f, *and* the solution in which v is instantiated by the constant function $f(a)$. This is an example of a 2^{nd}-*order-unification* problem. Note that there is no equational theory in the background here, but the relevant equality is not simply syntactic equality, either: some information about how functions behave is built into the theory. In particular, in 2^{nd}-order unification it is customary to take function variables as ranging over the functions definable in the lambda-calculus, and to take equality between functions to be axiomatized by the (β) and (η) equations of the lambda-calculus.

We are naturally led to combine these generalizations, allowing function variables and postulating certain equations between first-order terms as axioms. Naturally enough, this is called 2^{nd}-*order E-unification.*

Certain variations are significant in practice. One must often consider *simultaneous* unification problems, in which we seek a solution to a set of equations. Finally, it is sometimes important to consider "one-sided" unification, in which only one, designated, term in a pair is eligible for instantiation. This is the *matching* problem. Needless to say, each of these variations can be considered in an equational and/or a 2^{nd}-order framework.

Clearly both E-unification and 2^{nd}-order-unification are harder than syntactic unification, in an intuitive sense. One way to make this precise is to observe that syntactic unification is easily decidable, while each of E-unification and 2^{nd}-order-unification are undecidable in general (see [BaSi93] for a discussion). But beyond this the situation is not so straightforward.

In this paper we present some results which help to clarify the relationships among these paradigms, typically comparing them with respect to decidability. To derive these results it has turned out to be sufficient to look at equational theories that are generated by finite convergent term-rewriting systems in which only unary function symbols appear. A term-rewriting system of this form can easily be interpreted as a *string-rewriting system*, and we will make heavy use of string-rewriting terminology.

Summary of results

Let F denote a signature in which all function symbols occurring have arity 1, and let $\mathcal{E}$ be a set of equations which do *not* involve any individual constants, and in each of which the same individual variable occurs. Thus, $\mathcal{E}$ expresses some universal relationships among the functions in F (such as "f and g commute," or "f is idempotent"). Such equations may conveniently be presented as equations between strings.

A model $\mathcal{M}$ for $\mathcal{E}$ consists of a set M and a collection $F = \{f_i \mid i \in I\}$ of unary functions over M. In such a model we may ask whether an equation $s = t$ has a solution, when the variables range over the individuals and (perhaps) the functions of $\mathcal{M}$. But there are two natural choices for what we mean by "the functions of $\mathcal{M}$." One is to consider the functions which are lambda-definable from the elements of F — this leads to the full 2^{nd}-order E-unification problem. Another point of view would be to take function variables as denoting

only functions that are generated (by composition) from the interpretations of the elements of F. Here we are working in the monoid of functions over M that is generated by $\mathcal{M}$. This latter situation is precisely the *word unification* problem for $\mathcal{E}$, when $\mathcal{E}$ is treated simply as a set of string-equations.

So the same set of equations $\mathcal{E}$ defines three different notions of unification: the 1^{st}-order E-unification problem, the 2^{nd}-order E-unification problem, and the word unification problem. Of course we may also consider the analogous *matching* problems, and for each of these the *simultaneous* versions.

Two problems are said to be *(recursively) independent* if there exist theories $\mathcal{E}$ for which the one problem is decidable while the other problem is undecidable, and conversely. We show that the equational unification problem is independent of:

- the equational matching problem (Section 3),
- the word matching problem (Section 4), and
- the (simultaneous) 2^{nd}-order equational matching problem (Section 5).

In addition, we prove that the simultaneous E-unification problem is decidable for theories $\mathcal{E}$ which are defined by finite, monadic, and confluent string-rewriting systems (Section 3), and that for arbitrary $\mathcal{E}$ as above the (single-pair) 2^{nd}-order E-matching problem is recursively reducible to the (1^{st}-order) E-matching problem (Theorem 5.1). Finally, we prove that for finite string-rewriting systems in general, the 2^{nd}-order equational unification problem reduces to the word unification problem (Theorem 6.5), but that, on the other hand, there exists a finite, length-reducing, and confluent string-rewriting system for which the 2^{nd}-order equational unification problem is decidable, while the word unification problem is undecidable for this system (Theorem 6.6).

The paper is organized as follows. In Section 2 some basic definitions are given, and the terminology used is introduced. In Section 3 the 1^{st}-order equational matching and unification problems for finite, convergent string-rewriting systems are considered, and in Section 4 these problems are related to the equational word matching and unification problems. In Section 5 the 2^{nd}-order equational matching and unification problems are defined, and the above-mentioned results on the 2^{nd}-order equational matching problem are derived. Finally in Section 6 the 2^{nd}-order equational unification problem is related to the equational word unification problem. The paper closes with a short summary. In an appendix we discuss the 2^{nd}-order equational matching and unification problems for string-rewriting systems in the case that also function variables of arity larger than one are taken into account. As it will turn out these function variables do not make the problems considered more difficult. This justifies the definitions of 2^{nd}-order equational matching and unification that are used in the main body of the paper.

2 Preliminaries

Here we present the basic definitions concerning term-rewriting systems, equational unification, and string-rewriting systems that we will need throughout the paper. We keep the definitions given to a minimum – more information and discussion of the notions introduced can be found in the literature. For term-rewriting systems our main reference is Dershowitz and Jouannaud [DeJo90], for equational unification it is Baader and Siekmann [BaSi93], and for string-rewriting systems it is Book and Otto [BoOt93].

Let F be a finite set of function symbols, each $f \in F$ having a fixed arity $\alpha(f) \in \mathbf{N}$, and let X be a countably infinite set of variables. As usual, function symbols of arity 0 will be called constants. Then $T = T(F, X)$ denotes the set of *terms* generated by F and X.

A term $t \in T(F,X)$ can be seen as a finite ordered tree, the leaves of which are labeled with variables or constants and the internal nodes of which are labeled with function symbols of positive arity such that the outdegree of an internal node equals the arity of its label. Thus, a *position* within a term can be represented – in Dewey decimal notation – as the sequence of positive integers which describes the path from the root to that position. Accordingly, the set $O(t)$ of *occurrences* of the term t is the set of sequences of positive integers describing the positions in t. The length of the longest of these sequences is called the *depth* of the term t, which is denoted as $\mathrm{depth}(t)$, and the number of sequences in $O(t)$ is the *size* of t, denoted as $\mathrm{size}(t)$. For example, if $t = f(g(a), h(x))$, where f is a binary function symbol (that is, $\alpha(f) = 2$), g and h are unary function symbols (that is, $\alpha(g) = \alpha(h) = 1$), a is a constant, and x is a variable, then $O(t) = \{\varepsilon, 1, 2, 1.1, 2.1\}$, $\mathrm{depth}(t) = 2$, and $\mathrm{size}(t) = 5$. For $p \in O(t)$, $t|_p$ denotes the subterm of t at occurrence p. If s is another term, then $t[s]_p$ denotes the term that is obtained by replacing the subterm of t at occurrence p by the term s. For a term $t \in T(F,X)$, $\mathrm{Var}(t)$ denotes the set of variables that have occurrences in t. If no variable occurs more than once in t, then t is called a *linear* term.

A *substitution* is a mapping $\sigma : X \to T(F,X)$ such that $\sigma(x) = x$ holds for almost all variables x. It can uniquely be extended to a morphism $\sigma : T(F,X) \to T(F,X)$.

A *term-rewriting system* R is a (finite) set of *rules* $\{\ell_i \to r_i \mid i \in I\}$, where ℓ_i and r_i are terms from $T(F,X)$ such that $\mathrm{Var}(r_i) \subseteq \mathrm{Var}(\ell_i)$ $(i \in I)$.

A term t is *reducible* modulo R if there is a rule $\ell \to r$ in R, an occurrence $p \in O(t)$, and a substitution σ such that $\sigma(\ell) = t|_p$. The term $t[\sigma(r)]_p$ is the result of *reducing* t by $\ell \to r$ at p. By $\to_R$ we denote the *single-step reduction relation* defined by the term-rewriting system R. Its reflexive and transitive closure $\to_R^*$ is the *reduction relation* induced by R. A term t is said to be in *normal form* or *irreducible* modulo R if no reduction can be applied to t. By $\mathrm{IRR}(R)$ we denote the set of all irreducible terms.

The equational theory that is associated with a term-rewriting system R is the congruence $=_R$ that is generated by the reduction relation $\to_R$, that is, it is the congruence $\leftrightarrow_R^* := (\to_R \cup \leftarrow_R)^*$.

A term-rewriting system R is said to be *noetherian* if there are no infinite sequences of reductions, that is, for each term t, each sequence $t \to_R t_1 \to_R \cdots$ is finite. It is *locally confluent* if, for all terms s, t, and u, $s \to_R t$ and $s \to_R u$ imply that, for some term w, $t \to_R^* w$ and $u \to_R^* w$. It is *confluent* if, for all terms s, t, and u, $s \to_R^* t$ and $s \to_R^* u$ imply that, for some term w, $t \to_R^* w$ and $u \to_R^* w$. Finally, R is called *convergent* (or *complete*) if it is both noetherian and confluent. In this case each term has a unique normal form with respect to R, that is, for each term t, there exists one and only one irreducible term t_0 such that $t =_R t_0$. In addition, from t the term t_0 can be determined effectively by reduction. The system R is *depth-reducing* if $\mathrm{depth}(\ell) > \mathrm{depth}(r)$ holds for each rule $\ell \to r$ of R. It is *linear* if all the terms occurring as the left-hand side or the right-hand side of a rule of R are linear.

Let R be a term-rewriting system on $T(F,X)$. Two terms $s, t \in T(F,X)$ are said to be *unifiable modulo* R if there exists a substitution σ such that $\sigma(s) =_R \sigma(t)$ holds. The substitution σ is then called an *R-unifier* of s and t. We say that there exists an *R-match* from s onto t if there exists a substitution σ such that $\sigma(s) =_R t$.

As indicated in the introduction our main results will be concerned with term-rewriting systems that only involve function symbols of arity one. In fact, this class of term-rewriting systems is essentially just the class of *string-rewriting systems*.

Let Σ be an alphabet and Σ^* be the set of all strings over Σ including the empty string λ. For $w \in \Sigma^*$, $|\, w \,|$ denotes the length of w. Obviously, Σ^* is in one-to-one correspondence to the set of terms $T(\Sigma, \{x\})$, where each letter from Σ is simply interpreted as a unary function symbol.

A string-rewriting system S on an alphabet Σ is a (finite) set of pairs of strings from Σ^*. A pair (ℓ, r) is often referred to as a *rule*. Usually, the rule (ℓ, r) will be denoted as $(\ell \to r)$. For a rule $(\ell \to r)$, ℓ is said to be its *left-hand side* and r its *right-hand side*. The set of all left-hand sides of a string-rewriting system S is denoted by domain(S), and the set of its right-hand sides is denoted by range(S). Under the isomorphism between Σ^* and $T(\Sigma, \{x\})$ the string-rewriting system S corresponds to the term-rewriting system $R_S := \{\ell(x) \to r(x) \mid (\ell \to r) \in S\}$. Thus, the notions of being noetherian, locally confluent, confluent, and convergent immediately carry over to string-rewriting systems.

We close this section by introducing some additional notation that we will only need for the case of string-rewriting systems.

Let S be a string-rewriting system on Σ. For $u \in \Sigma^*$, $[u]_S$ denotes the congruence class $[u]_S = \{w \in \Sigma^* \mid u \longleftrightarrow_S^* w\}$, and for a language $L \subseteq \Sigma^*$, $[L]_S = \bigcup_{u \in L} [u]_S$.

A string-rewriting system S is called *length-reducing* if each rule $(\ell \to r)$ of S satisfies $|\ell| > |r|$. Observe that S is length-reducing if and only if the term-rewriting system R_S is depth-reducing. It is called *monadic* if it is length-reducing, and if range$(S) \subseteq \Sigma \cup \{\lambda\}$, that is, if the right-hand side of each rule of S is a single letter or the empty string. Finally, it is called *special* if it is length-reducing, and if range$(S) = \{\lambda\}$.

A string-rewriting system S is called *interreduced* if range$(S) \subset$ IRR(S), and if $\ell \in$ IRR$(S \setminus \{\ell \to r\})$ holds for each rule $(\ell \to r)$ of S. For each finite convergent system an equivalent finite convergent system can be computed effectively such that the latter system is also interreduced. Here two systems are called *equivalent* if they generate the same congruence relation. Therefore, we can always assume in the following that the finite convergent systems considered are interreduced. Such systems are called *canonical*.

3 Equational matching and unification problems

Let Σ be a finite alphabet, let S be a string-rewriting system on Σ, and let $V := \{v_i \mid i \in \mathbf{N}\}$ be a set of string variables such that $\Sigma \cap V = \emptyset$. We consider existential sentences of the following form:

$$\exists v_1, \ldots, v_n : g_1 \sim h_1 \text{ and } \ldots \text{ and } g_m \sim h_m,$$

where $g_i, h_i \in (\Sigma \cup \{v_1, \ldots, v_n\})^*$, $i = 1, \ldots, m$. We say that this sentence has a *solution for* S if there exists a mapping $\phi : \{v_1, \ldots, v_n\} \to \Sigma^*$ such that $\phi(g_i) \leftrightarrow_S^* \phi(h_i)$ holds for all $i = 1, \ldots, m$. Here ϕ is extended to $(\Sigma \cup \{v_1, \ldots, v_n\})^*$ in the obvious way. By varying the syntactic form of the existential sentences considered we can define various equational matching and unification problems for S.

First of all, if each symbol $a \in \Sigma$ is interpreted as a unary function symbol $a(.)$, and if S is interpreted as the term-rewriting system R_S on the signature $F_\Sigma := \{a(.) \mid a \in \Sigma\}$, then the resulting equational matching and unification problems for S can be defined as usual. Using existential sentences they can be expressed as follows.

(1.) The **equational (1st-order) matching problem** for S:

INSTANCE : Two strings $g, h \in \Sigma^*$.
QUESTION : Does the sentence "$\exists v : gv \sim h$" have a solution for S?

(2.) The **equational (1st-order) unification problem** for S:

INSTANCE : Two strings $g, h \in \Sigma^*$.
QUESTION : Does "$\exists v_1, v_2 : gv_1 \sim hv_2$" have a solution for S?

Here it is also possible that the two variables v_1 and v_2 coincide, that is, the question could be whether "$\exists v : gv \sim hv$" has a solution.

By considering more than one pair of strings at a time, we obtain the simultaneous versions of the above problems.

(3.) The **simultaneous equational (1^{st}-order) matching problem** for S:

INSTANCE : Some pairs $(g_1, h_1), \ldots, (g_m, h_m) \in \Sigma^* \times \Sigma^*$.
QUESTION : Does "$\exists v : g_1 v \sim h_1$ and ... and $g_m v \sim h_m$" have a solution for S?

(4.) The **simultaneous equational (1^{st}-order) unification problem** for S:

INSTANCE : Some pairs $(g_1, h_1), \ldots, (g_m, h_m) \in \Sigma^* \times \Sigma^*$.
QUESTION : Does "$\exists v_1, \ldots, v_k : g_1 v_{i_1} \sim h_1 v_{j_1}$ and ... and $g_m v_{i_m} \sim h_m v_{j_m}$" have a solution for S, where $i_1, \ldots, i_m, j_1, \ldots, j_m \in \{1, \ldots, k\}$, and $k \geq 1$?

Recall from [NaOt90] that the simultaneous E-matching and E-unification problems are in general more difficult than their non-simultaneous counterparts.

On the other hand, we have the "classical" word matching and unification problems, where each symbol $a \in \Sigma$ is considered as a constant, and where an additional binary function symbol ("concatenation") is used that is associative. These problems can be stated as follows.

(5.) The **word matching problem** for S:

INSTANCE : Two strings $g \in (\Sigma \cup V)^*$ and $h \in \Sigma^*$.
QUESTION : Does "$\exists v_1, \ldots, v_k : g \sim h$" have a solution for S, where $\{v_1, \ldots, v_k\} = \{v \in V \mid |g|_v > 0\}$?

(6.) The **word unification problem** for S:

INSTANCE : Two strings $g, h \in (\Sigma \cup V)^*$.
QUESTION : Does "$\exists v_1, \ldots, v_k : g \sim h$" have a solution for S, where $\{v_1, \ldots, v_k\} = \{v \in V \mid |g|_v + |h|_v > 0\}$?

Here $|w|_v$ denotes the v-length of the string w, that is, the number of occurrences of the symbol v in w.

Again, by considering more than one pair of strings at a time, we obtain the simultaneous versions of these problems. While for the empty system the simultaneous versions of these problems are reducible to the non-simultaneous versions [Pec81], it is not known whether this also holds for non-empty systems.

Recall that word unification generalizes 1^{st}-order unification and specializes 2^{nd}-order unification in that function variables are allowed to be instantiated, but only by functions definable explicitly from F. Concerning these problems the following results are known.

Proposition 3.1

(a) The equational matching problem is decidable in polynomial time for finite, monadic, and confluent string-rewriting systems [BoOt93].

(b) The word unification problem is decidable for the empty system $S = \emptyset$ [Mak77].

(c) There is a finite, monadic, and confluent system for which the word matching problem is undecidable [BoOt93].

(d) There is a finite, special, and confluent system for which the word unification problem is undecidable [Ott95].

In fact, as shown recently by Narendran and Otto [NaOt96] there even exists a finite, special, and confluent system for which the word matching problem is undecidable. Observe that neither the equational unification problem nor the simultaneous equational matching problem can be expressed by linear sentences in the sense of Book [Boo83, BoOt93].

We now turn to the relationship between the E-matching problem and the E-unification problem for finite convergent string-rewriting systems.

Theorem 3.2 *There exists a finite, length-reducing, and confluent string-rewriting system S_1 such that*

(a) the E-matching problem for S_1 is decidable, while

(b) the E-unification problem for S_1 is undecidable.

Proof. In [NaOt90] a finite, length-reducing, confluent string-rewriting system $T_2(S)$ on the alphabet $\Delta = \{a, b, a_1, b_1, c_1, c_2, c, d_1, d_2, d, e_2, \ldots, e_k, f_2, \ldots, f_k, g_2, \ldots, g_k, h, h_1, h_2, \$, \not\epsilon, \S\}$ is constructed from some finite set $P = \{(y_i, z_i) \mid i = 2, \ldots, k\} \subset \{a, b\}^+ \times \{a, b\}^+$ such that the E-unification problem for $T_2(S)$ is undecidable ([NaOt90], Theorem 3.4). We take $S_1 := T_2(S)$, and we verify that the E-matching problem for S_1 is decidable. Recall from [NaOt90] that S_1 contains the following rules:

$$
\left.
\begin{array}{lcl}
e_i c \rho(y_i) & \to & c \\
e_i d \rho(z_i) & \to & d \\
c_1 f_i g_i & \to & e_i c_1 \\
d_1 f_i g_i & \to & e_i d_1
\end{array}
\right\} \; i = 2, \ldots, k
$$

$$
\begin{array}{lcl}
c_1 h & \to & c \\
d_1 h & \to & d \\
\$c & \to & c_2 \\
\$d & \to & d_2 \\
a_1 c_2 a & \to & c_2 \\
b_1 c_2 b & \to & c_2 \\
a_1 d_2 a & \to & d_2 \\
b_1 d_2 b & \to & d_2 \\
h_1 c_2 \not\epsilon & \to & \S \\
h_2 d_2 \not\epsilon & \to & \S
\end{array}
$$

where ρ denotes the function reversal. First, we need the following technical result.

Claim 1. There exists a deterministic pushdown automaton (dpda) $B(S_1)$ which, when given a string $w \in \Delta^*$ as input, halts with the irreducible descendant of w mod S_1 on its pushdown store.

Proof. Let $u \in \mathrm{IRR}(S_1)$ and $v \in \Delta^*$ be such that

$$u = u_1 u_2,\ v = v_2 v_1, \text{ and } (u_2 v_2 \to r) \in S_1.$$

Since u is irreducible, $v_2 \neq \lambda$. Now $uv = u_1 u_2 v_2 v_1 \to_{S_1} u_1 r v_1$, and from the form of the rules of S_1 we see that either u_1 ends with an occurrence of the symbol \$ and $r \in \{c, d\}$, or $u_1 r$ is irreducible. In the former case $u_1 r = u_3 \$ c$ or $u_1 r = u_3 \$ d$, which reduces in one step to the irreducible string $u_3 c_2$ or $u_3 d_2$, respectively. Consider a left-most reduction

$$w = w_0 \to_{S_1} w_1 \to_{S_1} \ldots \to_{S_1} w_m \in \mathrm{IRR}(S_1).$$

Then, for $i = 0, 1, \ldots, m-1$, w_i can be factored as $w_i = u_i \ell_i v_i$ such that $u_i \in \mathrm{IRR}(S_1)$, $(\ell_i \to r_i) \in S_1$, and $w_i = u_i \ell_i v_i \to_{S_1} u_i r_i v_i = w_{i+1} = u_{i+1} \ell_{i+1} v_{i+1}$. The observation above implies that either $u_i = u_{i+1}\$$ and $r_i \in \{c, d\}$ or $|v_{i+1}| < |v_i|$. Using the technique of

Theorem 2.2.9 of [BoOt93], a deterministic two-stack machine $B(S_1)$ can be built that, on input $w \in \Delta^*$, computes the irreducible descendant w_0 of w mod S_1 by simulating the left-most reduction from w to w_0. Because of the above, stack 2 of $B(S_1)$ can be implemented as a read-only input tape. Thus, $B(S_1)$ is in fact a dpda. □

From Claim 1 we immediately obtain the following result.

Claim 2. If $A \subseteq \mathrm{IRR}(S_1)$ is a regular set, then $[A]_{S_1}$ is a deterministic context-free language. Given a finite-state acceptor for A, a dpda for $[A]_{S_1}$ can be constructed effectively.

In particular, for each $w \in \Delta^*$, $[w]_{S_1}$ is a deterministic context-free language, and from w a dpda for $[w]_{S_1}$ can be constructed effectively. Based on this observation we can now establish the following result.

Claim 3. The E-matching problem for S_1 is decidable.

Proof. Let $g, h \in \Delta^*$. There exists a string $w \in \Delta^*$ such that $gw \leftrightarrow^*_{S_1} h$ if and only if the language $g \cdot \Delta^* \cap [h]_{S_1}$ is nonempty. By Claim 2 $[h]_{S_1}$ is deterministic context-free, and hence, so is the language $g \cdot \Delta^* \cap [h]_{S_1}$. Since a dpda can be constructed for this language from g and h, we can determine whether or not it is empty. □

Thus S_1 has all the required properties. This completes the proof of Theorem 3.2. □

On the other hand, we have the following result.

Theorem 3.3 *There exists a finite, length-reducing, and confluent string-rewriting system S_2 such that*

(a) the E-matching problem for S_2 is undecidable, while

(b) the E-unification problem for S_2 is decidable.

Proof. The E-matching problem for a string-rewriting system S is called the right-divisibility problem for S in [BoOt93]. There an example of a finite, length-reducing, and confluent string-rewriting system S_0 on a finite alphabet Σ_0 is given such that this problem is undecidable for S_0 ([BoOt93], Corollary 5.2.5). We construct the system S_2 as an extension of S_0. Let $\Sigma = \Sigma_0 \cup \{Z\}$, where Z is a new symbol, and let S_2 denote the following string-rewriting system on Σ:

$$S_2 = S_0 \cup \{aZ \to Z \mid a \in \Sigma\}.$$

Then S_2 is a finite length-reducing system, and since S_0 is confluent, and Z is a new symbol, it is easily seen that S_2 is confluent too. For all $g, h \in \Sigma^*$, $gZ \to^*_{S_2} Z$ and $hZ \to^*_{S_2} Z$, and hence, the existential sentence "$\exists v_1, v_2 : gv_1 \sim hv_2$" always has the solution $\{v_1 \leftarrow Z, v_2 \leftarrow Z\}$. Thus, the E-unification problem for S_2 is decidable.

On the other hand, let $g, h \in \Sigma_0^*$. There exists a string $w \in \Sigma^*$ such that $gw \leftrightarrow^*_{S_2} h$ if and only if there exists a string $w \in \Sigma_0^*$ such that $gw \leftrightarrow^*_{S_0} h$. However, this problem is undecidable by the choice of S_0. Thus, the E-matching problem for S_2 is undecidable. □

Theorems 3.2 and 3.3 show that for the class of finite, length-reducing, and confluent string-rewriting systems, the E-matching problem and the E-unification problem are independent. At first, this result, which for the case of term-rewriting systems has been observed by Bürckert [Bür89], may appear as a surprise, since it is generally expected that a unification problem is at least as hard as the corresponding matching problem. But the proof of Theorem 3.3 reveals the reason for this unexpected behavior: when we interpret the system S_2 as a term-rewriting system $R_{S_2} = \{\ell(x) \to r(x) \mid (\ell \to r) \in S_2\}$ over the set of unary function symbols $F_\Sigma = \{a(.) \mid a \in \Sigma\}$, then we see that we have no constants and, hence, no ground

terms that could be used to describe instances of the E-matching problem as special instances of the E-unification problem mod R_{S_2}. As soon as we have an additional, uninterpreted function symbol, that is, a letter that does not occur in any rule of the string-rewriting system under consideration, the situation changes. This is the contents of the following theorem.

Theorem 3.4 *Let S be a string-rewriting system on some alphabet Σ. If there exists a letter in Σ that does not occur in any rule of S, then the E-matching problem for S is reducible to the E-unification problem for S.*

Proof. We want to give a reduction from the E-matching problem for S to the E-unification problem for S. So let "$\exists v : gv \sim h$" be an instance of the E-matching problem for S. Assume that there exists a letter $\$ \in \Sigma$ that does not occur in any rule of S. We can write g and h, respectively, as $g = g_0\$g_1\$\ldots\$g_n$ and $h = h_0\$h_1\$\ldots\$h_m$, where $g_0, \ldots, g_n, h_0, \ldots, h_m \in (\Sigma \setminus \{\$\})^*$ and $n, m \geq 0$. If $n > m$, then the given instance of the E-matching problem has no solution for S, since, for all $w \in \Sigma^*$, $|gw|_\$ \geq |g|_\$ = n > m = |h|_\$$, and applications of rules of S do not change the number of occurrences of the symbol \$. So assume that $n \leq m$.

Claim 1. If there exists a string $w \in \Sigma^*$ such that $gw \leftrightarrow_S^* h$, then there exist strings $u, v \in \Sigma^*$ such that $gu \leftrightarrow_S^* h\$v$.

Proof. Choose $u = w\$$ and $v = \lambda$. Then $gu = gw\$ \leftrightarrow_S^* h\$ = h\$v$. □

Claim 2. If there exist strings $u, v \in \Sigma^*$ such that $gu \leftrightarrow_S^* h\$v$, then there exists a string $w \in \Sigma^*$ such that $gw \leftrightarrow_S^* h$.

Proof. We can write u and v as $u = u_0\$u_1\$\ldots\$u_p$ and $v = v_0\$v_1\$\ldots\$v_q$, where $u_0, \ldots, u_p$, $v_0, \ldots, v_q \in (\Sigma \setminus \{\$\})^*$ and $p, q \geq 0$. By the hypothesis of Claim 2, $g_0\$g_1\$\ldots\$g_n u_0\$u_1\$\ldots\$u_p$ $= gu \leftrightarrow_S^* h\$v = h_0\$h_1\$\ldots\$h_m\$v_0\$v_1\$\ldots\$v_q$. Since no rule of S contains an occurrence of the symbol \$, this implies that $n + p = |gu|_\$ = |h\$v|_\$ = m + 1 + q$, and $g_i \leftrightarrow_S^* h_i$ for all $i = 0, 1, \ldots, n-1$, $g_n u_0 \leftrightarrow_S^* h_n$, $u_i \leftrightarrow_S^* h_{n+i}$ for all $i = 1, \ldots, m-n$, and $u_i \leftrightarrow_S^* v_{q-p+i}$ for all $i = m-n+1, \ldots, p$. Take $w := u_0\$\ldots\$u_{m-n}$. Then $gw = g_0\$g_1\$\ldots\$g_n u_0\$u_1\$\ldots\$u_{m-n} \leftrightarrow_S^*$ $h_0\$h_1\$\ldots\$h_n\$h_{n+1}\$\ldots\$h_m = h$. □

Thus, the given instance of the E-matching problem for S has a solution if and only if the instance "$\exists u, v : gu \sim h\$v$" of the E-unification problem for S has a solution. This completes the proof of Theorem 3.4. □

Analogously, the corresponding result can be shown for the simultaneous E-matching and E-unification problems.

Corollary 3.5 *Let S be a string-rewriting system on some alphabet Σ. If there exists a letter in Σ that does not occur in any rule of S, then the simultaneous E-matching problem for S is reducible to the simultaneous E-unification problem for S.*

We close this section with some positive results on the (simultaneous) E-matching and E-unification problems for the class of finite, monadic, and confluent string-rewriting systems. These results are based on a careful analysis of the reduction process with respect to these systems.

Let S be a finite, monadic, and confluent string-rewriting system on Σ. For $g, h \in \mathrm{IRR}(S)$ we want to characterize those strings $u, w \in \mathrm{IRR}(S)$ that satisfy the congruence $gu \leftrightarrow_S^* hw$. Since S is confluent, $gu \leftrightarrow_S^* hw$ implies that gu and hw have a common descendant mod S. Thus, if u and w are irreducible strings satisfying the congruence above, then S being monadic implies that there exist $a, b \in \Sigma \cup \{\lambda\}$ and factorizations $g = g_1 g_2$, $u = u_2 u_1$, $h = h_1 h_2$, and $w = w_2 w_1$ such that the following three conditions are satisfied:

(i) $g_2 u_2 \to_S^* a$,

(ii) $h_2 w_2 \to_S^* b$, and

(iii) $g_1 a u_1 = h_1 b w_1 \in \mathrm{IRR}(S)$.

Based on the two factorizations of $g_1 a u_1$ we obtain three cases:

(1.) If $|g_1| = |h_1|$, then $g_1 = h_1$, $a = b$ or one of a, b is the empty string λ, and $a u_1 = b w_1$.

(2.) If $|g_1| > |h_1|$, then $h_1 b$ is a prefix of g_1, that is, $g_1 = h_1 b h_3$, and $w_1 = h_3 a u_1$.

(3.) If $|g_1| < |h_1|$, then $g_1 a$ is a prefix of h_1, that is, $h_1 = g_1 a g_3$, and $u_1 = g_3 b w_1$.

For $x \in \mathrm{IRR}(S)$ and $a \in \Sigma \cup \{\lambda\}$, let $\mathrm{RF}(x,a)$ denote the set $\mathrm{RF}(x,a) = \{y \in \mathrm{IRR}(S) \mid xy \to_S^* a\}$. Then $\mathrm{RF}(x,a)$ is a regular language, and from x and a, a nondeterministic finite-state acceptor (nfa) can be constructed in polynomial time for $\mathrm{RF}(x,a)$ ([Ott86], Theorem 5.1). Using sets of this form we get the following characterization.

Lemma 3.6 *Let S be a finite, monadic, and confluent string-rewriting system on Σ, and let $g, h \in \mathrm{IRR}(S)$. Then the following two statements are equivalent:*

(a) $\exists u, w \in \Sigma^* : gu \leftrightarrow_S^* hw$.

(b) $\exists g_1, g_2, h_1, h_2, y \in \Sigma^*, \exists a, b \in \Sigma \cup \{\lambda\} : g = g_1 g_2$ *and* $h = h_1 h_2$ *and* $\mathrm{RF}(g_2, a) \neq \emptyset \neq \mathrm{RF}(h_2, b)$ *and (i)* $g_1 = h_1$ *and* ($a = b$ *or* $\lambda \in \{a, b\}$) *or (ii)* $g_1 = h_1 b y$ *or (iii)* $h_1 = g_1 a y$.

Proof. From the observation above we see that (a) implies (b). On the other hand, it is easily verified that (b) implies (a). □

For $g, h \in \mathrm{IRR}(S)$ we are also interested in those strings $w \in \mathrm{IRR}(S)$ that satisfy the congruence $gw \leftrightarrow_S^* hw$. If $g, h, w \in \mathrm{IRR}(S)$ satisfy $gw \leftrightarrow_S^* hw$, then there exist factorizations $g = g_1 g_2$, $h = h_1 h_2$, $w = w_2 w_1 = w_4 w_3$ and $a, b \in \Sigma \cup \{\lambda\}$ such that the following conditions are satisfied:

(i) $g_2 w_2 \to_S^* a$,

(ii) $h_2 w_4 \to_S^* b$, and

(iii) $g_1 a w_1 = h_1 b w_3 \in \mathrm{IRR}(S)$.

Based on the two factorizations of w we get the following three cases:

(1.) If $|w_1| = |w_3|$, then $w_2 = w_4$ and $w_1 = w_3$. Hence, $g_1 a = h_1 b$, and $w_2 = w_4 \in \mathrm{RF}(g_2, a) \cap \mathrm{RF}(h_2, b)$.

(2.) If $|w_1| > |w_3|$, then $w_4 = w_2 w_5$ and $w_1 = w_5 w_3$ for some string $w_5 \in \Sigma^*$. Hence, $h_1 b = g_1 a w_5$ and $h_2 w_4 = h_2 w_2 w_5 \to_S^* b$. Thus, $w_2 \in \mathrm{RF}(g_2, a)$ and $w_4 \in \mathrm{RF}(h_2, b) \cap \mathrm{RF}(g_2, a) \cdot w_5$.

(3.) If $|w_1| < |w_3|$, then $w_2 = w_4 w_5$ and $w_3 = w_5 w_1$ for some string $w_5 \in \Sigma^*$. Hence, $g_1 a = h_1 b w_5$ and $g_2 w_2 = g_2 w_4 w_5 \to_S^* a$. Thus, $w_4 \in \mathrm{RF}(h_2, b)$ and $w_2 \in \mathrm{RF}(g_2, a) \cap \mathrm{RF}(h_2, b) \cdot w_5$.

Hence, we get the following characterization.

Lemma 3.7 *Let S be a finite, monadic, and confluent string-rewriting system on Σ, and let $g, h \in \mathrm{IRR}(S)$. Then the following two statements are equivalent:*

(a) $\exists w \in \Sigma^* : gw \leftrightarrow_S^* hw$.

(b) $\exists g_1, g_2, h_1, h_2, y \in \Sigma^*, \exists a, b \in \Sigma \cup \{\lambda\} : g = g_1 g_2$ *and* $h = h_1 h_2$ *and*

(i) $g_1 a = h_1 b$ *and* $\mathrm{RF}(g_2, a) \cap \mathrm{RF}(h_2, b) \neq \emptyset$ *or*

(ii) $h_1 b = g_1 a y$ *and* $\mathrm{RF}(g_2, a) \cdot y \cap \mathrm{RF}(h_2, b) \neq \emptyset$ *or*

(iii) $g_1 a = h_1 b y$ *and* $\mathrm{RF}(g_2, a) \cap \mathrm{RF}(h_2, b) \cdot y \neq \emptyset$.

For $g, h \in \mathrm{IRR}(S)$, there are only $|g| \cdot |h|$ factorizations of the form $g = g_1 g_2$ and $h = h_1 h_2$. Thus, Lemma 3.6 and Lemma 3.7 give the following result.

Theorem 3.8 *The E-unification problem is decidable in polynomial time for each finite, monadic, and confluent string-rewriting system.*

Finally, we turn to the simultaneous E-matching and E-unification problems for finite, monadic, and confluent string-rewriting systems. Recall from Proposition 3.1(a) that the (non-simultaneous) E-matching problem is decidable in polynomial time for each finite, monadic, and confluent string-rewriting system.

Theorem 3.9 *The simultaneous E-matching problem is decidable for each finite, monadic, and confluent string-rewriting system.*

Proof. Let S be a finite, monadic, and confluent string-rewriting system on Σ, let $v_1, \ldots, v_k$ be some variables, and let $(g_1, h_1), \ldots, (g_m, h_m) \in \Sigma^* \times \Sigma^*$. We want to check whether or not the existential sentence

$$\exists v_1, \ldots, v_k : g_1 v_{i_1} \sim h_1 \text{ and } \ldots \text{ and } g_m v_{i_m} \sim h_m$$

has a solution for S. Here $v_{i_1}, \ldots, v_{i_m} \in \{v_1, \ldots, v_k\}$.

Since each subexpression $g_j v_{i_j} \sim h_j$ contains only a single occurrence of a single variable, we can rearrange the sentence above into a conjunction of sentences of the form

$$\exists v : g_1 v \sim h_1 \text{ and } \ldots \text{ and } g_k v \sim h_k.$$

Thus, it suffices to deal with the special case of having a single variable only. For $i = 1, \ldots, k$, if $w \in \Sigma^*$ satisfies $g_i w \leftrightarrow_S^* h_i$, then there exist $a \in \Sigma \cup \{\lambda\}$ and factorizations $g_i = f_1 f_2$, $w = w_2 w_1$, and $h_i = f_1 a w_1$ such that $f_2 w_2 \rightarrow_S^* a$, since we can assume without loss of generality that g_i and h_i are irreducible mod S, $i = 1, \ldots, k$. Thus, $\mathrm{Sol}(g_i, h_i) := \left(\bigcup_{\substack{g_i = f_1 f_2 \\ a \in \Sigma \cup \{\lambda\} \\ h_i = f_1 a w_1}} \mathrm{RF}(f_2, a) \cdot w_1 \right) \cap \mathrm{IRR}(S)$ is the set of all possible solutions of the subexpression $g_i v \sim h_i$. Obviously, $\mathrm{Sol}(g_i, h_i)$ is a regular set, and from g_i, h_i, and S an nfa can be constructed for $\mathrm{Sol}(g_i, h_i)$ in polynomial time. Now the existential sentence

$$\exists v : g_1 v \sim h_1 \text{ and } \ldots \text{ and } g_k v \sim h_k$$

has a solution if and only if $\bigcap_{i=1,\ldots,k} \mathrm{Sol}(g_i, h_i) \neq \emptyset$. Thus, the simultaneous E-matching problem for S is decidable. □

Observe that the regular set $\bigcap_{i=1,\ldots,k} \mathrm{Sol}(g_i, h_i)$ describes the set of all (irreducible) solutions of the existential sentence "$\exists v : g_1 v \sim h_1$ and ... and $g_k v \sim h_k$." Hence, the set of (irreducible) solutions of the more general existential sentence

$$\exists v_1, \ldots, v_k : g_1 v_{i_1} \sim h_1 \text{ and } \ldots \text{ and } g_m v_{i_m} \sim h_m$$

is given by a collection of regular sets $R_1, \ldots, R_k$ such that each k-tuple $(w_1, \ldots, w_k) \in R_1 \times \ldots \times R_k$ gives a solution under the assignment $v_1 \leftarrow w_1, \ldots, v_k \leftarrow w_k$.

Unfortunately, the exact degree of complexity of the simultaneous E-matching problem for finite, monadic, and confluent string-rewriting systems is still unknown. However, it is unlikely that this problem is solvable in polynomial time. An indication for this is the following completeness result for the uniform version of this problem, where the finite, monadic, and confluent string-rewriting system considered is taken as a part of the problem instance.

Theorem 3.10 *The uniform version of the simultaneous E-matching problem for finite, monadic, and confluent string-rewriting systems is* PSPACE*-complete.*

Proof. From the proof of Theorem 3.9 we can easily conclude that the uniform version of the simultaneous E-matching problem for finite, monadic, and confluent string-rewriting systems is solvable in polynomial space. Thus, in order to establish the announced completeness result it suffices to give a polynomial-time reduction from some known PSPACE-complete problem to the uniform version of the simultaneous E-matching problem for finite, monadic, and confluent string-rewriting systems. Here we use the finite state automata intersection problem, which is defined as follows:

INSTANCE	:	A sequence $A_1, A_2, \ldots, A_m$ of deterministic finite state acceptors having the same input alphabet Σ.
QUESTION	:	Is there a string $w \in \Sigma^*$ that is accepted by each of the A_i, $i = 1, 2, \ldots, m$?

It is known that this problem is PSPACE-complete (see, e.g., [GaJo79]).

Let $A = (Q, \Sigma, q_0, F, \delta)$ be a deterministic finite state acceptor (dfa), and let $L(A) = B \subseteq \Sigma^*$ be the language accepted by A. From A we construct a finite monadic string-rewriting system $S(A)$ on the alphabet $\Delta := Q \cup \Sigma \cup \{\$, \#\}$ as follows:

$$\begin{array}{lcll} q_i a & \to & q_j & \text{if} \quad \delta(q_i, a) = q_j, \\ q_i \# & \to & \$ & \text{if} \quad q_i \in F. \end{array}$$

Then $S(A)$ is also confluent, and for all $w \in \Delta^*$, $q_0 w \leftrightarrow^*_{S(A)} \$$ if and only if $q_0 w \to^*_{S(A)} \$$ if and only if $w \in B \cdot \{\#\}$.

By performing this construction for m dfa's $A_i = (Q_i, \Sigma, q_{0,i}, F_i, \delta_i)$, $i = 1, \ldots, m$, in parallel, where the state sets Q_i are pairwise disjoint, we obtain the finite, monadic, and confluent string-rewriting system $S := \bigcup_{1 \le i \le m} S(A_i)$ on the alphabet $\Gamma := (\bigcup_{1 \le i \le m} Q_i) \cup \Sigma \cup \{\#, \$\}$. The existential sentence "$\exists v : q_{0,1} v \sim \$$ and ... and $q_{0,m} v \sim \$$" has a solution for S if and only if $L(A_1) \cdot \{\#\} \cap \ldots \cap L(A_m) \cdot \{\#\} \neq \emptyset$ if and only if $L(A_1) \cap \ldots \cap L(A_m) \neq \emptyset$. Thus, the uniform version of the simultaneous E-matching problem for finite, monadic, and confluent string-rewriting systems is indeed PSPACE-complete. □

The simultaneous E-unification problem for finite, monadic, and confluent string-rewriting systems seems to be even more difficult. However, we still have the following decidability result.

Theorem 3.11 *The simultaneous E-unification problem is decidable for each finite, monadic, and confluent string-rewriting system.*

Proof. Let S be a finite, monadic, and confluent string-rewriting system on Σ, and let $(g_1, h_1), \ldots, (g_m, h_m) \in \Sigma^* \times \Sigma^*$ be some pairs of irreducible strings. We consider the following existential sentence:

$$\exists v_1, \ldots, v_k : g_1 v_{i_1} \sim h_1 v_{j_1} \text{ and } \ldots \text{ and } g_m v_{i_m} \sim h_m v_{j_m},$$

where $i_1, \ldots, i_m, j_1, \ldots, j_m \in \{1, \ldots, k\}$. We look at each subexpression $g_\ell v_{i_\ell} \sim h_\ell v_{j_\ell}$ in turn. We distinguish between the two cases that $v_{i_\ell} = v_{j_\ell}$ and that $v_{i_\ell} \neq v_{j_\ell}$.

(i) If $v_{i_\ell} = v_{j_\ell}$, then we see from Lemma 3.7 and its proof that the set

$$\mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell}) = \Big(\bigcup_{\substack{g', g'', h', h'' \\ a, b \in \Sigma \cup \{\lambda\} \\ g_\ell = g'g'' \\ h_\ell = h'h'' \\ g'a = h'b}} (\mathrm{RF}(g'', a) \cap \mathrm{RF}(h'', b)) \cdot \Sigma^*$$

$$\cup \bigcup_{\substack{g', g'', h', h'', y \\ g_\ell = g'g'' \\ h_\ell = h'h'' \\ a, b \in \Sigma \cup \{\lambda\} \\ h'b = g'ay}} (\mathrm{RF}(g'', a) \cdot y \cap \mathrm{RF}(h'', b)) \cdot \Sigma^* \cup \bigcup_{\substack{g', g'', h', h'', y \\ g_\ell = g'g'' \\ h_\ell = h'h'' \\ a, b \in \Sigma \cup \{\lambda\} \\ g'a = h'by}} (\mathrm{RF}(g'', a) \cap \mathrm{RF}(h'', b) \cdot y) \cdot \Sigma^*)$$

$\cap \mathrm{IRR}(S)$

is the set of all irreducible solutions of the subsentence

$$\exists v_{i_\ell} : g_\ell v_{i_\ell} \sim h_\ell v_{i_\ell}.$$

(ii) If $v_{i_\ell} \neq v_{j_\ell}$, then we see from Lemma 3.6 and its proof that the set

$$\mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell}, v_{j_\ell}) = \{w \# z \# x \mid (w, z) \in \bigcup_{\substack{g', g'', h'' \\ g_\ell = g'g'' \\ h_\ell = g'h'' \\ a \in \Sigma \cup \{\lambda\}}} (\mathrm{RF}(g'', a) \times \mathrm{RF}(h'', a))$$

$$\cup \bigcup_{\substack{g', g'', h'' \\ g_\ell = g'g'' \\ h_\ell = g'h'' \\ a \in \Sigma}} (\mathrm{RF}(g'', a) \times \mathrm{RF}(h'', \lambda) \cdot a) \cup \bigcup_{\substack{g', g'', h'' \\ g_\ell = g'g'' \\ h_\ell = g'h'' \\ b \in \Sigma}} (\mathrm{RF}(g'', \lambda) \cdot b \times \mathrm{RF}(h'', b))$$

$$\cup \bigcup_{\substack{g', g'', h', h'', y \\ g_\ell = g'g'' \\ h_\ell = h'h'' \\ a, b \in \Sigma \cup \{\lambda\} \\ g' = h'by}} (\mathrm{RF}(g'', a) \times \mathrm{RF}(h'', b) \cdot ya) \cup \bigcup_{\substack{g', g'', h', h'', y \\ g_\ell = g'g'' \\ h_\ell = h'h'' \\ a, b \in \Sigma \cup \{\lambda\} \\ h' = g'ay}} (\mathrm{RF}(g'', a) \cdot yb \times \mathrm{RF}(h'', b))$$

and $x \in \Sigma^*$ such that $wx, zx \in \mathrm{IRR}(S)\}$

describes all irreducible solutions $\{v_{i_\ell} \leftarrow wx, v_{j_\ell} \leftarrow zx\}$ of the subsentence

$$\exists v_{i_\ell}, v_{j_\ell} : g_\ell v_{i_\ell} \sim h_\ell v_{j_\ell},$$

where $\#$ is an additional symbol not in Σ.

The sets $\mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell})$ and $\mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell}, v_{j_\ell})$ $(\ell = 1, \ldots, m)$ are regular, and nfa's for them can be constructed effectively.

If $w_1, \ldots, w_k \in \mathrm{IRR}(S)$ give a solution for the existential sentence above under the substitution $\{v_1 \leftarrow w_1, \ldots, v_k \leftarrow w_k\}$, then the following conditions are satisfied for $\ell = 1, \ldots, m$:

(i) if $v_{i_\ell} = v_{j_\ell} = v_r$, then $w_r \in \mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell})$, and

(ii) if $v_{i_\ell} = v_r \neq v_s = v_{j_\ell}$, then there exists $y\#z\#x \in \mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell}, v_{j_\ell})$ such that $w_r = yx$ and $w_s = zx$ hold, that is, there exist $y, z, x \in \Sigma^*$ and $s \in \mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell}, v_{j_\ell})$ such that

- $s = y\#z\#x$,
- $w_r = yx$, and
- $w_s = zx$.

Conversely, if $w_1, \ldots, w_k \in \mathrm{IRR}(S)$ satisfy conditions (i) and (ii) above for all $\ell = 1, \ldots, m$, then the substitution $\{v_1 \leftarrow w_1, \ldots, v_k \leftarrow w_k\}$ is obviously a solution for the given existential sentence.

For all $\ell = 1, \ldots, m$, we construct some word equations over $(\Sigma \cup \{\#\})$ as follows:

(i) if $v_{i_\ell} = v_{j_\ell} = v_r$, then we take the equation $x_r = y_r$, and for y_r we choose the regular set $\mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell})$ as domain;

(ii) if $v_{i_\ell} = v_r \neq v_s = v_{j_\ell}$, then we take the equations $x_r = yx$ and $x_s = zx$ and $s = y\#z\#x$, where x, y, and z have domain Σ^*, while s has domain $\mathrm{Sol}(g_\ell, h_\ell, v_{i_\ell}, v_{j_\ell})$.

Finally, for the variables $x_1, \ldots, x_k$ we take the domain Σ^*. Then the existential sentence given has a solution if and only if these word equations simultaneously have a solution over $\Sigma \cup \{\#\}$, where each variable takes a value from its domain. Since all these domains are regular sets, this problem is decidable by Schulz's extension of Makanin's result on the solvability of word equations in free semigroups [Sch90]. Thus, simultaneous E-unification is decidable for S. □

Actually, Schulz only considers a single word equation. However, using an additional new letter, the finite system of word equations constructed in the proof above can be combined into a single word equation that has exactly the same solutions as the finite system.

We close this section with a short example.

Example 3.12 *Let $\Sigma = \{a, b, c, d, e, f\}$ and $S = \{ca \to \lambda, eb \to \lambda, da^2 \to d, df \to \lambda\}$. Then S is a finite monadic system that is confluent and interreduced. Let us consider the following instance of the simultaneous E-unification problem for S:*

$(*)$ $\exists v_1, v_2, v_3 : c^n v_1 \sim e^n v_2$ and $dv_1 \sim e^n v_3$.

Then

$$\begin{aligned}\mathrm{Sol}(c^n, e^n, v_1, v_2) \quad = \quad & \{a^n\#b^n\#w \mid w \in \mathrm{IRR}(S)\} \\ & \cup \{a^{n-i}\#b^n c^i\#w \mid w \in \mathrm{IRR}(S), w \notin a \cdot \Sigma^*, i \in \{1, \ldots, n\}\} \\ & \cup \{a^n e^i\#b^{n-i}\#w \mid w \in \mathrm{IRR}(S), w \notin b \cdot \Sigma^*, i \in \{1, \ldots, n\}\}\end{aligned}$$

and

$$\begin{aligned}\mathrm{Sol}(d, e^n, v_1, v_3) \quad = \quad & \{a^{2j} f \# b^n \# w \mid w \in \mathrm{IRR}(S),\ j \geq 0\} \\ & \cup \{a^{2j} \# b^n d \# w \mid w \in \mathrm{IRR}(S), w \notin \{a^2, f\} \cdot \Sigma^*,\ j \geq 0\} \\ & \cup \{a^{2j} f e^i \# b^{n-i} \# w \mid w \in \mathrm{IRR}(S), w \notin b \cdot \Sigma^*, j \geq 0,\ i \in \{1, \ldots, n\}\}.\end{aligned}$$

From $(*)$ we obtain the following word equations and domains:

$$\exists x_1, x_2, x_3 \in \Sigma^*, \exists x, y, z, x', y', z' \in \Sigma^*, \exists s_1 \in \mathrm{Sol}(c^n, e^n, v_1, v_2), \exists s_2 \in \mathrm{Sol}(d, e^n, v_1, v_3) :$$

$$\begin{aligned} s_1 &= x\#y\#z, \\ s_2 &= x'\#y'\#z', \\ x_1 &= xz, \\ x_2 &= yz, \\ x_1 &= x'z', \\ x_3 &= y'z'. \end{aligned}$$

For example, $s_1 = a^2\#b^n c^{n-2}\#$ and $s_2 = a^2\#b^n d\#$ give $x_1 = a^2$, $x_2 = b^n c^{n-2}$ and $x_3 = b^n d$, which yields a solution for $(*)$.

The finite system of word equations is finally combined into a single word equation over the enlarged alphabet $\Delta = \Sigma \cup \{\#, \$\}$:

$$s_1\$s_2\$x_1\$x_2\$x_1\$x_3 = x\#y\#z\$x'\#y'\#z'\$xz\$yz\$x'z'\$y'z'.$$

The regular constraints for the variables $s_1, s_2, x_1, x_2, x_3, x, y, z, x', y', z'$ remain the same as before. □

Note that even though *unifiability* is decidable, complete sets of unifiers may be infinite. In fact, there are convergent monadic systems whose unification is of type *nullary,* that is, minimal, complete sets of unifiers may not exist in some cases. A simple example is the string-rewriting system $\{ab \to a, ac \to a, ad \to a, bd \to \lambda, cd \to \lambda\}$ and the terms ax and ay, that is, $a(x)$ and $a(y)$ when viewed as terms over unary function symbols.

4 Equational unification and word matching

Theorem 3.11 and Proposition 3.1(c) yield the following separation result.

Corollary 4.1 *There exists a finite, monadic, and confluent string-rewriting system S such that*

(a) the simultaneous E-unification problem for S is decidable, while

(b) the word matching problem for S is undecidable.

For future reference, in particular in Section 5, we now restate in short the construction of a finite, monadic, and confluent string-rewriting system that has the properties stated in Corollary 4.1.

As described in [NaOt90] there exists a finite set $P := \{(y_i, z_i) \mid i = 2, \ldots, k\}$ of pairs of non-empty strings $y_i, z_i \in \{a, b\}^+$ $(i = 2, \ldots, k)$ such that the following version of the modified Post Correspondence Problem (MPCP) is undecidable:

INSTANCE	:	Two strings $y_1, z_1 \in \{a, b\}^+$.
QUESTION	:	Is there a sequence of integers $i_1, \ldots, i_m \in \{2, \ldots, k\}$ such that $y_1 y_{i_1} \ldots y_{i_m} = z_1 z_{i_1} \ldots z_{i_m}$?

If such a sequence of integers exists, then it is called a *solution* of the instance $\{(y_1, z_1)\} \cup P$ of the MPCP. We say that MPCP(y_1, z_1) has a solution to express the fact that such a solution exists.

From the above finite set P we now construct a string-rewriting system $T_m(P)$ on the alphabet $\Sigma = \{a, b, c_2, \ldots, c_k, \not{c}, \$, \#\}$. This system consists of the following monadic rules:

$$\left.\begin{array}{rcl} y_i \$ c_i & \to & \$ \\ z_i \not{c} c_i & \to & \not{c} \end{array}\right\} \; i = 2, \ldots, k.$$

It is easily verified that $T_m(P)$ has the following properties.

Lemma 4.2

(a) The string-rewriting system $T_m(P)$ is finite, monadic, confluent, and interreduced.

(b) For all $y_1, z_1 \in \{a, b\}^+$, the following two statements are equivalent:

(i) MPCP(y_1, z_1) has a solution.

(ii) There are strings $g, h \in \Sigma^*$ such that $g\$h\#g\not{c}h \leftrightarrow^*_{T_m(P)} y_1\$\#z_1\not{c}$.

Proof.

(a) obvious.

(b) If $i_1, \ldots, i_m \in \{2, \ldots, k\}$ is a solution for MPCP(y_1, z_1), then take $g = y_1 y_{i_1} \ldots y_{i_m}$ $(= z_1 z_{i_1} \ldots z_{i_m})$ and $h = c_{i_m} \ldots c_{i_1}$. Conversely, if $g, h \in \Sigma^*$ satisfy $g\$h\#g\not{c}h \leftrightarrow^*_{T_m(P)}$ $y_1\$\#z_1\not{c}$, then $g\$h\#g\not{c}h \to^*_{T_m(P)} y_1\$\#z_1\not{c}$, since $y_1\$\#z_1\not{c}$ is irreducible, and $T_m(P)$ is confluent. From the form of the rules of $T_m(P)$ it then follows that $g = y_1 y_{i_1} \ldots y_{i_m}$ and $h = c_{i_m} \ldots c_{i_1}$ for some indices $i_1, \ldots, i_m \in \{2, \ldots, k\}$, and that also $g = z_1 z_{i_1} \ldots z_{i_m}$. Thus, $i_1, \ldots, i_m$ is a solution for MPCP(y_1, z_1). □

From the choice of the set P and from Theorem 3.11 we thus see that the system $T_m(P)$ has the properties stated in Corollary 4.1. We will also need the following technical property of the system $T_m(P)$.

Since $T_m(P)$ is monadic, we know that, for all $w, u, z \in \mathrm{IRR}(T_m(P))$, if $wu \to^+_{T_m(P)} z$, then w, u, and z can be factored as $w = w_1 w_2$, $u = u_2 u_1$, and $z = w_1 d u_1$, where $w_2 u_2 \to^*_{T_m(P)} \ell$ and $(\ell \to d) \in T_m(P)$. The particular form of the rules of $T_m(P)$ shows that $d = \$$ or $d = \not{c}$, and that $w_2 u_2 = y_{i_m} \ldots y_{i_1} \$ c_{i_1} \ldots c_{i_m}$ or $w_2 u_2 = z_{i_m} \ldots z_{i_1} \not{c} c_{i_1} \ldots c_{i_m}$ for some $m \geq 1$ and some indices $i_1, \ldots, i_m \in \{2, \ldots, k\}$. Since w and u are irreducible, we see that $y_{i_m} \ldots y_{i_2}$ (or $z_{i_m} \ldots z_{i_2}$) is a proper prefix of w_2, and that $c_{i_1} \ldots c_{i_m}$ is a suffix of u_2, that is, $w_2 = y_{i_m} \ldots y_{i_2} y'$ $(z_{i_m} \ldots z_{i_2} z')$ for some nonempty prefix y' of y_{i_1} (z' of z_{i_1}) and $u_2 = y''\$c_{i_1} \ldots c_{i_m}$ $(z''\not{c}c_{i_1} \ldots c_{i_m})$, where $y_{i_1} = y'y''$ $(z_{i_1} = z'z'')$, or $w_2 = y_{i_m} \ldots y_{i_1}\$$ $(z_{i_m} \ldots z_{i_1}\not{c})$ and $u_2 = c_{i_1} \ldots c_{i_m}$. Since the strings $y_2, \ldots, y_k$ and $z_2, \ldots, z_k$ are nonempty, this means that w has only finitely many suffixes of this particular form. Thus, there are only finitely many different choices for the string u_2. This observation is expressed by the following lemma.

Lemma 4.3 *For each irreducible string $w \in \Sigma^*$, a finite set $\mathrm{RM}(w)$ of irreducible strings exists such that the following conditions are satisfied:*

(1.) For each string $u \in \mathrm{RM}(w)$ there exist a factorization $w = w_1 w_2$ and some $d \in \Sigma \cup \{\lambda\}$ such that $wu = w_1 w_2 u \to^*_{T_m(P)} w_1 d$, and $w_1 d$ is irreducible.

(2.) Whenever $y, z \in \mathrm{IRR}(T_m(P))$ satisfy $wy \to^*_{T_m(P)} z$, there exist an element $u \in \mathrm{RM}(w)$ and a string $y' \in \Sigma^*$ such that $y = uy'$ and $z = w_1 d y'$, where $w = w_1 w_2$ and $d \in \Sigma \cup \{\lambda\}$ correspond to $u \in \mathrm{RM}(w)$ according to (1.).

From the discussion above it is easily seen that the finite set $\mathrm{RM}(w)$ can be determined effectively from w.

Obviously, the word matching problem is a special case of the word unification problem. The following result shows that in general the latter is more difficult than the former.

Theorem 4.4 *There exists a finite, length-reducing, and confluent string-rewriting system for which the word matching problem is decidable, while the word unification problem is undecidable.*

Proof. As above let $P = \{(y_i, z_i) \mid i = 2, \ldots, k\} \subseteq \{a,b\}^+ \times \{a,b\}^+$ be chosen such that the problem $\mathrm{MPCP}(y_1, z_1)$ $(y_1, z_1 \in \{a,b\}^+)$ is undecidable. Let $\Sigma = \{a, b, e_2, \ldots, e_k, c, d, \$, \S\}$, let $n = max\{|y_i|, |z_i| \mid i = 2, \ldots, k\} + 1$, and let $T_c(P)$ denote the following string-rewriting system on Σ:

$$\left.\begin{array}{lcl} ce_i^n & \to & y_i c \\ de_i^n & \to & z_i d \end{array}\right\} \; i = 2, \ldots, k$$
$$\begin{array}{lcl} c\$ & \to & \S \\ d\$ & \to & \S \end{array}$$

Then $T_c(P)$ is length-reducing, and since it has no nontrivial critical pairs, it is also confluent.

Claim 1. For all $y_1, z_1 \in \{a,b\}^+$, the following two statements are equivalent:

(i) $\mathrm{MPCP}(y_1, z_1)$ has a solution.

(ii) The sentence "$\exists v : y_1 cv \sim z_1 dv$" has a solution mod $T_c(P)$.

Proof. If $i_1, \ldots, i_m \in \{2, \ldots, k\}$ is a solution for $\mathrm{MPCP}(y_1, z_1)$, then $v = e_{i_1}^n \ldots e_{i_m}^n \$$ satisfies $y_1 cv = y_1 ce_{i_1}^n \ldots e_{i_m}^n \$ \to^*_{T_c(P)} y_1 y_{i_1} \ldots y_{i_m} \S = z_1 z_{i_1} \ldots z_{i_m} \S \leftarrow^*_{T_c(P)} z_1 de_{i_1}^n \ldots e_{i_m}^n \$ = z_1 dv$. Conversely, if $v \in \Sigma^*$ satisfies $y_1 cv \leftrightarrow^*_{T_c(P)} z_1 dv$, then we see from the form of the rules of $T_c(P)$ that $v = e_{i_1}^n \ldots e_{i_m}^n \$ v_1$ for some $i_1, \ldots, i_m \in \{2, \ldots, k\}$ and $v_1 \in \Sigma^*$. Here we assume that v is irreducible. Then $y_1 cv = y_1 ce_{i_1}^n \ldots e_{i_m}^n \$ v_1 \to^*_{T_c(P)} y_1 y_{i_1} \ldots y_{i_m} \S v_1$ and $z_1 dv = z_1 de_{i_1}^n \ldots e_{i_m}^n \$ v_1 \to^*_{T_c(P)} z_1 z_{i_1} \ldots z_{i_m} \S v_1$. Since $T_c(P)$ is confluent, and since the strings $y_1 y_{i_1} \ldots y_{i_m} \S v_1$ and $z_1 z_{i_1} \ldots z_{i_m} \S v_1$ are irreducible, they must coincide, and so, $y_1 y_{i_1} \ldots y_{i_m} = z_1 z_{i_1} \ldots z_{i_m}$. □

Since the problem $\mathrm{MPCP}(y_1, z_1)$ $(y_1, z_1 \in \{a,b\}^+)$ is undecidable in general, this means that the E-unification problem, and in particular, the word unification problem is undecidable for the system $T_c(P)$.

Claim 2. The word matching problem for $T_c(P)$ is decidable.

Proof. Consider the existential sentence

$$\exists v_1, \ldots, v_k : g_0 v_{i_1} g_1 \ldots v_{i_m} g_m \sim h,$$

where $g_0, g_1, \ldots, g_m, h \in \Sigma^*$ are irreducible, and $v_{i_1}, \ldots, v_{i_m} \in \{v_1, \ldots, v_k\}$. If this sentence has a solution $\{v_i \leftarrow w_i \mid i = 1, \ldots, k\}$, then $g_0 w_{i_1} g_1 \ldots w_{i_m} g_m \to^*_{T_c(P)} h$. Let Γ denote the subalphabet $\Gamma = \{a, b, \S\}$ of Σ. From the form of the rules of $T_c(P)$ we see that $u \to_{T_c(P)} w$

implies that $|u|_\Gamma < |w|_\Gamma$, where $|u|_\Gamma$ denotes the Γ-*length* of u, that is, the number of occurrences of symbols from Γ in u. Hence, the set of ancestors $\{u \in \Sigma^* \mid u \rightarrow^*_{T_c(P)} h\}$ of h mod $T_c(P)$ is finite. Thus, we can decide whether or not the existential sentence above has a solution mod $T_c(P)$. □□

The proof of Theorem 4.4 also shows the following.

Corollary 4.5 *There exists a finite, length-reducing, and confluent string-rewriting system for which the word matching problem is decidable, while the E-unification problem is undecidable.*

5 The 2nd-order E-matching problem

We finally turn to the 2nd-order equational matching problem. Let S be a string-rewriting system on Σ; as before we will interpret S as embodying an equational theory of unary functions, and will consider unification and matching problems modulo S. The sense in which the problems in this section are "higher-order" is this: substitutions may replace function variables by any term definable in the lambda-calculus, using only unary second-order variables; and there is an additional, "built-in" notion of equality between terms, that is generated by the familiar (β) and (η) axioms. However, we will adopt a notation in which, as explained below, $(\beta\eta)$ equality can be handled *implicitly.*

An observation on notation: in the logic literature systems with one-place functions and predicates are called "monadic." Since this has nothing to do with the notion of "monadic string-rewriting system," we will avoid confusion and consistently use the term "unary" to delimit function-arity.

In order to incorporate the intuitions and technical results on string-rewriting systems, it will be convenient to use a concrete syntax for terms and substitutions which does *not* use explicit abstraction. Indeed, the notation we use is essentially that of Goldfarb [Gol81] and Farmer [Far88]. We wish to emphasize that the difference between our presentation and currently standard presentations of second-order logic are purely superficial, and so we detail the correspondence below.

So assume as before that we have a set Σ of symbols, whose elements will be treated as (unary) function constants, and a set V of (unary) function variables. From the point of view of lambda-calculus, these are the atoms of functional type $\iota \rightarrow \iota$, where ι is the base type. In order to build terms of base type ι, we need to have some atoms of base type, so we assume our language includes a non-empty set Δ of individual constants and a set X of individual variables. We may now build lambda-terms by closing the atoms under application $f(t)$ and abstraction over a variable $\Lambda x.t$ (here t is any term, and we use the capital Λ here to avoid confusion with the earlier use of λ to denote the empty string). It is easy to check that (under the constraints imposed by the "unary" character of the atoms) the terms of base type which are in normal form under $\beta\eta$-reduction are precisely those of the form $(a_1(a_2 \cdots (a_n(c) \cdots)))$, where the a_i are atoms from $\Sigma \cup V$ and c is an atom from $\Delta \cup X$. (It is well-known that any term in our calculus is reducible to one in normal form).

Hence, the 2nd-order terms of base type in $T_2(F_\Sigma, V, X)$ are in 1-to-1 correspondence with the strings in the language $(\Sigma \cup V)^* \cdot (\Delta \cup X)$. For the remainder of this paper we will interpret such strings as lambda-terms, without additional comment.

Now return to the string-rewriting system S on Σ. We have seen that S can be considered as a term-rewriting system on the 1st-order terms $T(F_\Sigma, X)$, and we extend S to be a rewriting system on the second-order terms simply by treating the function variables V as free function symbols. Thus, for $s, t \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$ we have $s \rightarrow_S t$ if and only if $\exists g, h \in (\Sigma \cup V)^*, \exists d \in X \cup \Delta, \exists (\ell \rightarrow r) \in S : s = g\ell h(d)$ and $t = grh(d)$.

The problems of higher-order matching and unification differ from their 1st-order counterparts in the complexity of the substitutions allowed. A substitution may replace a function variable by any term of function-type; for example, if $u, v \in V$, $a, b \in \Sigma$, and $x \in X$, one might replace u by the term $\Lambda x.a(v(a(b(x))))$, which simply designates the function-composition more readily denoted by the string $avab$. In fact, it is easy to see that when Δ is empty, any closed term of type $\iota \to \iota$ is of the form $\Lambda x.a_1(a_2(\ldots(a_n(x))\ldots))$, where $a_1, \ldots, a_n \in \Sigma \cup V$ and $x \in X$. These are (once substituted into a term and β-reduced) just the strings from $(\Sigma \cup V)^*$. Hence, if Δ is empty, then word matching is essentially the same as second-order matching, and similarly for unification.

However, when Δ is non-empty, we have the second-order constant functions, that is, terms of the form $\Lambda x.a_1(a_2(\ldots(a_n(c))\ldots))$, where $a_1, \ldots, a_n \in \Sigma \cup V$ and $c \in \Delta$. For example, the function-term $\Lambda x.a(a(b(c)))$ denotes the constant function $f(x) = a(a(b(c)))$, and this function cannot be described as a composition of functions from $\Sigma \cup V$. The assumption that Δ is non-empty is the usual one in studies of the lambda-calculus. In this case, the difference between word matching and 2nd-order matching is dramatic: this is demonstrated by Theorem 5.1. In Section 6 we will see that the situation is similar, but more delicate, for unification.

Note that after *applying* a substitution to a base-type term, no matter what form the substitution takes, the resulting term will be Λ-free. But in order to manipulate the substitutions themselves we must have an extended notation.

To this end, let $\Box$ be an additional symbol that will be used as a "place holder." By interpreting $\Box$ as an additional individual constant we can form the set $T_2(F_\Sigma \cup \{\Box\}, V, X)$ of *extended 2nd-order terms*, which corresponds to the language $(\Sigma \cup V)^* \cdot (\Delta \cup \{\Box\} \cup X)$. A *2nd-order substitution* is a mapping $\phi : V \cup X \to (\Sigma \cup V)^* \cdot (\Delta \cup \{\Box\} \cup X) \cup V$ that satisfies the following three conditions:

(1) $\mathrm{dom}(\phi) = \{x \in V \cup X \mid \phi(x) \neq x\}$ is finite,

(2) $\phi(x) \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$ for all $x \in X$, and

(3) $\phi(v) \in (\Sigma \cup V)^* \cdot (\Delta \cup \{\Box\} \cup X)$ for all $v \in \mathrm{dom}(\phi) \cap V$.

The intended interpretation is that for $v \in V$, $\phi(v) \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$ means that the 1-place function variable v is to be replaced by a constant function (and $\phi(v)$ denotes the value obtained for any argument), while $\phi(v) = w(\Box)$ means that v will be replaced by the function of arity 1 that is defined by w.

Observe that the above definition is based on our assumption that function variables have arity 1. The theory can be developed along the same lines when admitting also function variables of arity larger than 1, but as shown in the appendix the same results regarding 2nd-order equational matching and unification are obtained for string-rewriting systems. Therefore, we restrict our attention here to the (notationally less complicated) case of admitting only function variables of arity 1.

With this in mind we may extend a 2nd-order substitution ϕ to a mapping

$$\phi_e : (\Sigma \cup V)^* \cdot (\Delta \cup X) \to (\Sigma \cup V)^* \cdot (\Delta \cup X)$$

as follows:

- if $g \in \Delta$, then $\phi_e(g) = g$,
- if $g \in X$, then $\phi_e(g) = \phi(g)$,
- if $g = ag_1$ for some $a \in \Sigma$, then $\phi_e(g) := a(\phi_e(g_1))$,
- if $g = vg_1$ for some $v \in V$ such that $v \notin \mathrm{dom}(\phi)$, then $\phi_e(g) = v(\phi_e(g_1))$,

– if $g = vg_1$ for some $v \in V \cap \mathrm{dom}(\phi)$ such that $\phi(v) \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$, then $\phi_e(g) = \phi(v)$, and

– if $g = vg_1$ for some $v \in V \cap \mathrm{dom}(\phi)$ such that $\phi(v) = w(\Box)$ for some $w \in (\Sigma \cup V)^*$, then $\phi_e(g) := w(\phi_e(g_1))$.

To simplify the notation we will denote the extension ϕ_e of ϕ simply by ϕ.

We may now consider second-order matching and unification modulo a string-rewriting system S. More precisely (in this paper) we are reasoning modulo an equational theory which has equations only between second-order terms with no individual constants. For example we may express that f is idempotent by: $ff = f$. To say that f is a left-inverse for g we would write: $fg = \lambda$.

The **2nd-order E-matching problem** for S is now defined as follows:

INSTANCE : Two strings $g, h \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$.
QUESTION : Is there a 2^{nd}-order substitution ϕ such that $\phi(g) \leftrightarrow_S^* h$?

Accordingly, the **2nd-order E-unification problem** for S is defined as:

INSTANCE : Two strings $g, h \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$.
QUESTION : Is there a 2^{nd}-order substitution ϕ such that $\phi(g) \leftrightarrow_S^* \phi(h)$?

The way the function variables are used here is similar to the way the string variables are used in the word matching and word unification problems. However, there are two differences. First of all, function variables may be replaced by terms (strings) that still contain variables. More importantly, when dealing with substitutions in the framework of the word matching and word unification problems, then a (string) variable is simply replaced by its value, that is, a string. In the framework of 2^{nd}-order substitutions, the situation is more complicated: if $\phi(v) = w(\Box)$, then $\phi(vg) = w(\phi(g))$, but if $\phi(v) = w \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$, then $\phi(vg) = w$. This has some surprising consequences as we will see.

Although 2^{nd}-order unification is undecidable in general, even for the empty theory [Gol81], it is decidable [Far88] whether two terms s, t in our "unary" language are 2^{nd}-order unifiable (in the empty equational theory).

Recall from Section 4 that there exists a finite, monadic, and confluent string-rewriting system for which the word matching problem is undecidable, while, on the other hand, the simultaneous E-matching and E-unification problems are decidable for each string-rewriting system of this form (Theorems 3.9 and 3.11). Such a situation cannot occur for the 2^{nd}-order E-matching problem, since the following result holds.

Theorem 5.1 *The 2^{nd}-order E-matching problem for a string-rewriting system S on Σ is effectively reducible to the (1^{st}-order) E-matching problem for S, where S is considered as a string-rewriting system on $\Sigma \cup V$.*

Proof. Let $g, h \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$. If $|g|_V = 0$, then $g = g_1 b$ for some $g_1 \in \Sigma^*$ and $b \in (\Delta \cup X)$. If $b \in X$, then we actually have an instance of the E-matching problem for the string-rewriting system S on $\Sigma \cup V$. If $b \in \Delta$, then there exists a substitution ϕ satisfying $\phi(g) \leftrightarrow_S^* h$ if and only if $h = h_1 b$ for some $h_1 \in \Sigma^*$ and $g_1 \leftrightarrow_S^* h_1$ holds. Exploiting the fact that the function variables in V are interpreted as free symbols for $\leftrightarrow_S^*$, this is equivalent to saying that $h = h_1 b$ for some $h_1 \in \Sigma^*$ and the existential sentence "$\exists v : g_1 uv \sim h_1 u$" has a solution mod S, where u is an arbitrary element of V (cf. the proof of Theorem 3.4). If $|g|_V > 0$, then $g = fug_1$ for some string $f \in \Sigma^*$ and some function variable $u \in V$. Let $h_1 \in (\Sigma \cup V)^*$ and $b \in (\Delta \cup X)$ such that $h = h_1 b$.

Claim. There exists a 2^{nd}-order substitution ϕ satisfying $\phi(g) \leftrightarrow_S^* h$ if and only if there is a string $w \in (\Sigma \cup V)^*$ such that $fw \leftrightarrow_S^* h_1$, that is, if and only if the existential sentence "$\exists v : fv \sim h_1$" has a solution mod S.

Proof. First, assume that ϕ is a 2^{nd}-order substitution that satisfies $\phi(g) \leftrightarrow_S^* h$. Since $g = fug_1$, we see that $\phi(g) = f\phi(ug_1)$. Let $w_1 := \phi(ug_1) \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$. Then w_1 can be factored as $w_1 = wd$ with $w \in (\Sigma \cup V)^*$ and $d \in \Delta \cup X$. Since $fwd = fw_1 = \phi(g) \leftrightarrow_S^* h = h_1 b$, we conclude that $b = d$ and $fw \leftrightarrow_S^* h_1$.

Conversely, assume that $w \in (\Sigma \cup V)^*$ satisfies $fw \leftrightarrow_S^* h_1$. Define a 2^{nd}-order substitution ϕ through $\phi(u) := wb$ and $\phi(u') := u'$ for all $u' \in (V \setminus \{u\}) \cup X$. Then $\phi(g) = \phi(fug_1) = f\phi(ug_1) = f\phi(u) = fwb \leftrightarrow_S^* h_1 b = h$. □

Thus, we have a reduction from the 2^{nd}-order E-matching problem for S (on Σ) to the (1^{st}-order) E-matching problem for S (on $\Sigma \cup V$). □

Because of Proposition 3.1(a) this yields the following immediate consequence.

Corollary 5.2 *For finite, monadic, and confluent string-rewriting systems the 2^{nd}-order E-matching problem is decidable in polynomial time.*

Thus, for the finite, monadic, and confluent string-rewriting system $T_m(P)$ of Section 4 we have the following situation:

(i) the simultaneous E-matching and E-unification problems are decidable,

(ii) the word matching and unification problems are undecidable, and

(iii) the 2^{nd}-order E-matching problem is decidable.

Also Theorem 3.2 yields the following.

Corollary 5.3 *There exists a finite, length-reducing, and confluent string-rewriting system S_1 such that*

(a) the 2^{nd}-order E-matching problem for S_1 is decidable, but

(b) the E-unification problem for S_1 is undecidable.

One consequence of Theorem 5.1 is the observation that in order to obtain equational theories (rewriting systems) for which the 2^{nd}-order E-matching problem is strictly more difficult than the (1^{st}-order) E-matching and E-unification problems, we must at least consider the *simultaneous* versions of these problems. We may achieve the same effect by adding a free binary function constant to the signature. Indeed, we return to one of the theories investigated in Section 4.

Let $T_m(P)$ be the finite, monadic, and confluent string-rewriting system on the alphabet $\Sigma = \{a, b, c_2, \ldots, c_k, \not\in, \$, \#\}$ from Section 4. From Σ we obtain a set of function symbols $F = \Sigma \cup \{c, f\}$, where each letter from Σ is interpreted as a unary function symbol, c is a new constant, and f is a new binary function symbol. By T_1 we denote the set of 1^{st}-order terms $T(F, X)$, and by T_2 we denote the set of 2^{nd}-order terms $T_2(F, V, X)$. (The modification to the definition needed to incorporate binary function constants should be obvious.) In the following the string-rewriting system $T_m(P)$ is interpreted as a term-rewriting system on T_1 as well as on T_2. In the latter case the function variables $v \in V$ are treated as free unary function symbols with respect to the reduction relation induced by $T_m(P)$. The congruence that $T_m(P)$ induced on T_1 and on T_2 will be written as $\Leftrightarrow^*$.

Theorem 5.4 *The 2^{nd}-order E-matching problem is undecidable for $T_m(P)$, when the system $T_m(P)$ is considered as a term-rewriting system on T_2.*

Proof. In Lemma 4.2 we have seen that the following restricted version of the word matching problem is undecidable for the string-rewriting system $T_m(P)$:

INSTANCE : Two strings $y_1, z_1 \in \{a,b\}^+$.
QUESTION : Do there exist strings $g, h \in \Sigma^*$ such that $g\$h\#g\not\!\!\$h \leftrightarrow^*_{T_m(P)} y_1\$\#z_1\not\!\!\$$ holds?

Here we will reduce this problem to the 2^{nd}-order E-matching problem for the term-rewriting system $T_m(P)$ on T_2.

Let $y_1, z_1 \in \{a,b\}^+$. We form the terms $s := f(v\$(x), v\not\!\!\$(x))$ and $t := f(y_1\$(c), z_1\not\!\!\$(c))$, where $x \in X$ and $v \in V$. Here $y_1\$(c)$ denotes the term that is built from the string $y_1\$$ by interpreting each letter as a unary function symbol, and accordingly for $z_1\not\!\!\$(c)$. We consider the instance (s,t) of the 2^{nd}-order E-matching problem for $T_m(P)$, that is, we ask whether there exists a 2^{nd}-order substitution $\phi : \{v,x\} \to T_2(F \cup \{\Box\}, V, X)$ satisfying $\phi(s) \Leftrightarrow^* t$. Here $\Leftrightarrow^*$ denotes the congruence relation on T_2 that is induced by the term-rewriting system $T_m(P)$. Observe that $t \in T_1$, and hence, $t' \in T_1$ holds for all $t' \in T_2$ satisfying $t \Leftrightarrow^* t'$. Thus, if a 2^{nd}-order substitution ϕ is to satisfy $\phi(s) \Leftrightarrow^* t$, then $\phi(x) \in T_1$ and $\phi(v) \in T(F \cup \{\Box\}, X)$ must hold necessarily.

Claim. The following two statements are equivalent:

(i) There exists a 2^{nd}-order substitution ϕ such that $\phi(s) \Leftrightarrow^* t$.

(ii) There exist strings $g, h \in \Sigma^*$ such that $g\$h\#g\not\!\!\$h \leftrightarrow^*_{T_m(P)} y_1\$\#z_1\not\!\!\$$.

Proof of claim. (ii) $\Rightarrow$ (i): Let $g, h \in \Sigma^*$ such that $g\$h\#g\not\!\!\$h \leftrightarrow^*_{T_m(P)} y_1\$\#z_1\not\!\!\$$. We define a 2^{nd}-order substitution ϕ through $\phi(x) := h(c)$ and $\phi(v) := g(\Box)$. Then $\phi(s) = \phi(f(v\$(x), v\not\!\!\$(x))) = f(g\$h(c), g\not\!\!\$h(c)) \Leftrightarrow^* f(y_1\$(c), z_1\not\!\!\$(c)) = t$.

(i) $\Rightarrow$ (ii): Let $\phi : \{v,x\} \to T_2(F \cup \{\Box\}, V, X)$ be a 2^{nd}-order substitution such that $\phi(s) \Leftrightarrow^* t$. As observed above this means that $\phi(x) \in T_1$ and $\phi(v) \in T(F \cup \{\Box\}, X)$.

First assume that $\phi(v) = t_1 \in T_1$. Then $\phi(s) = \phi(f(v\$(x), v\not\!\!\$(x))) = f(t_1, t_1) \Leftrightarrow^* t = f(y_1\$(c), z_1\not\!\!\$(c))$, which implies that $y_1\$(c) \Leftrightarrow^* t_1 \Leftrightarrow^* z_1\not\!\!\$(c)$, since no rule of $T_m(P)$ contains an occurrence of the function symbol f. This in turn means that $y_1\$ \leftrightarrow^*_{T_m(P)} z_1\not\!\!\$$, which contradicts the fact that $T_m(P)$ is confluent, since $y_1\$$ and $z_1\not\!\!\$$ are both irreducible mod $T_m(P)$. Thus, $\phi(v) \notin T_1$, that is, $\phi(v)$ contains some occurrences of the special "place holder" $\Box$. Since $|s|_f = 1 = |t|_f$, and since f is a free function symbol for $T_m(P)$, $|\phi(v)|_f$ must be 0, that is, $\phi(v) = g(\Box)$ for some $g \in \Sigma^*$. Let $\phi(x) := h_1 \in T_1$. Again it follows that $|h_1|_f = 0$, that is, $h_1 = h(d)$ for some $h \in \Sigma^*$ and $d \in X \cup \{c\}$. Hence, $f(y_1\$(c), z_1\not\!\!\$(c)) = t \Leftrightarrow^* \phi(s) = f(g\$h(d), g\not\!\!\$h(d))$, and so $g\$h(d) \Leftrightarrow^* y_1\$(c)$ and $g\not\!\!\$h(d) \Leftrightarrow^* z_1\not\!\!\$(c)$. Thus $d = c$, $g\$h \leftrightarrow^*_{T_m(P)} y_1\$$, and $g\not\!\!\$h \leftrightarrow^*_{T_m(P)} z_1\not\!\!\$$. $\Box$

From the choice of the system $T_m(P)$ we conclude that the 2^{nd}-order E-matching problem for $T_m(P)$ is undecidable, when $T_m(P)$ is considered as a term-rewriting system on T_2. $\Box$

In contrast to this undecidability result we now prove that the simultaneous E-unification problem is decidable for the term-rewriting system $T_m(P)$ on T_2. Actually, this result can be obtained as a consequence of Baader's and Schulz's result that general unification (where uninterpreted function symbols are allowed) is decidable for a theory $\mathcal{E}$ if elementary unification with linear constant restrictions (over free constants) is decidable for $\mathcal{E}$ [BaSc96]. However, in order to make this paper self-contained, we provide an elementary proof for the decidability

of the simultaneous E-unification problem for $T_m(P)$. For doing so we first have to establish some simple results on the congruence relation $\Leftrightarrow^*$ on T_2.

Since for the reduction process mod $T_m(P)$ the function variables from V are treated as free function symbols, we can restrict our attention to the 1^{st}-order terms in $T_1 = T(F, X)$. Hence, in what follows we consider $T_m(P)$ as a term-rewriting system on T_1.

Lemma 5.5 *Let $g = g_0(f(g_1, g_2))$ and $h = h_0(f(h_1, h_2))$ be terms from T_1 such that $g_0, h_0 \in \Sigma^*$. Then $g \Leftrightarrow^* h$ if and only if $g_0 \leftrightarrow^*_{T_m(P)} h_0$ and $g_i \Leftrightarrow^* h_i$ for $i = 1, 2$.*

Proof. "$\Leftarrow$": obvious.

"$\Rightarrow$": Assume that $g_0(f(g_1, g_2)) = g \Leftrightarrow^* h = h_0(f(h_1, h_2))$. Since no rule of $T_m(P)$ contains any occurrences of the binary function symbol f, we see that, whenever a rule $(\ell \to r) \in T_m(P)$ is applied to g, then ℓ is either a factor of g_0, or it is a subterm of g_1 or g_2. Thus, $g = g_0(f(g_1, g_2)) \Rightarrow^* w_0(f(w_1, w_2))$ for some irreducible term $w_0(f(w_1, w_2))$, and $h = h_0(f(h_1, h_2)) \Rightarrow^* w_0(f(w_1, w_2))$ as well, since also as a term-rewriting system on T_1, the system $T_m(P)$ is noetherian and confluent. Here $w_0 \in \Sigma^*$ satisfying $g_0 \to^*_{T_m(P)} w_0 \leftarrow^*_{T_m(P)} h_0$, and $g_i \Rightarrow^* w_i \Leftarrow^* h_i$ for $i = 1, 2$. This completes the proof of the lemma. □

Lemma 5.6 *If $g = g_0(f(g_1, g_2))$ and $h = h_0(x)$ are unifiable* mod $T_m(P)$*, where $g_0, h_0 \in \Sigma^*$ and $x \in X$, then the variable x does not occur in g, and $g_0 \leftrightarrow^*_{T_m(P)} h_0 w_0$ for some string $w_0 \in \Sigma^*$, that is, the mapping $x \mapsto w_0(f(g_1, g_2))$ is a match from h onto g for $T_m(P)$.*

Proof. Assume first that the variable x does occur in g, that is, $|g|_x > 0$. For each substitution ψ this gives $|\psi(h)|_f = |\psi(x)|_f < 1 + |\psi(x)|_f \leq |g_0(f(\psi(g_1), \psi(g_2)))|_f = |\psi(g)|_f$. On the other hand, if $\psi(h) \Leftrightarrow^* \psi(g)$, then $|\psi(h)|_f = |\psi(g)|_f$. Thus, if the variable x occurs in g, then the terms g and h are not unifiable mod $T_m(P)$.

Let φ be a unifier of g and h mod $T_m(P)$, that is, $\varphi(g) = g_0(f(\varphi(g_1), \varphi(g_2))) \Leftrightarrow^* h_0(\varphi(x)) = \varphi(h)$. Since $|\varphi(g)|_f > 0$ and $h_0 \in \Sigma^*$, we see that $|\varphi(x)|_f > 0$, that is, $\varphi(x) = w_0(f(w_1, w_2))$ for some string $w_0 \in \Sigma^*$. Now $g_0(f(\varphi(g_1), \varphi(g_2))) = \varphi(g) \Leftrightarrow^* \varphi(h) = h_0 w_0(f(w_1, w_2))$ implies that $g_0 \leftrightarrow^*_{T_m(P)} h_0 w_0$ and $\varphi(g_i) \Leftrightarrow^* w_i$, $i = 1, 2$, by Lemma 5.5. Thus, the mapping $\varphi_0 : x \mapsto w_0(f(g_1, g_2))$ satisfies $\varphi_0(h) = h_0 w_0(f(g_1, g_2)) \Leftrightarrow^* g_0(f(g_1, g_2)) = g$. □

Based on these two technical lemmas we now derive the announced decidability result.

Theorem 5.7 *The E-unification problem is decidable for the term-rewriting system $T_m(P)$.*

Proof. We prove this decidability result by reducing the E-unification problem for the term-rewriting system $T_m(P)$ to the simultaneous E-unification problem for the string-rewriting system $T_m(P)$, which is decidable by Theorem 3.11. Actually, we will consider the simultaneous version of the E-unification problem for the term-rewriting system $T_m(P)$, which, however, is equivalent to the nonsimultaneous variant because of the free binary function symbol f. So we will actually prove that the following problem is decidable:

INSTANCE	:	A finite sequence $(g_1, h_1), \ldots, (g_m, h_m) \in T_1 \times T_1$.
QUESTION	:	Does this sequence have a solution mod $T_m(P)$, that is, does there exist a substitution φ such that $\varphi(g_i) \Leftrightarrow^* \varphi(h_i)$ holds simultaneously for all $i = 1, \ldots, m$?

The proof will be done by noetherian induction. To this end we define a partial ordering $>$ on the set of all finite sequences from $T_1 \times T_1$ as the transitive closure of the following relation:

$$((g_1, h_1), \ldots, (g_m, h_m)) > ((g'_1, h'_1), \ldots, (g'_n, h'_n)) \text{ if and only if}$$

(i) the sequence $(g_1, h_1), \ldots, (g_m, h_m)$ contains strictly more different variables than the sequence $(g'_1, h'_1), \ldots, (g'_n, h'_n)$, that is, $\bigcup_{j=1}^{m} (\mathrm{Var}(g_j) \cup \mathrm{Var}(h_j)) \supsetneq \bigcup_{j=1}^{n} (\mathrm{Var}(g'_j) \cup \mathrm{Var}(h'_j))$,

or

(ii) there exist an index $i \in \{1, \ldots, m\}$ and a finite subset $J \subseteq \{1, \ldots, n\}$ such that $\{(g'_1, h'_1), \ldots, (g'_n, h'_n)\} = \{(g_j, h_j) \mid j = 1, \ldots, m,\ j \neq i\} \cup \{(g'_j, h'_j) \mid j \in J\}$ and, for each $j \in J$, g'_j is a proper subterm of g_i, and h'_j is a proper subterm of h_i, that is, the sequence $(g'_1, h'_1), \ldots, (g'_n, h'_n)$ is obtained from the sequence $(g_1, h_1), \ldots, (g_m, h_m)$ by replacing a pair (g_i, h_i) by finitely many pairs (g'_j, h'_j) consisting of proper subterms of g_i and h_i, respectively.

It is easily verified that the partial ordering $>$ is well-founded. Hence, we can use this partial ordering for the intended noetherian induction.

If the terms in the sequence $(g_1, h_1), \ldots, (g_m, h_m)$ do not contain any occurrences of the binary function symbol f, then it is essentially an instance of the simultaneous E-unification problem for the string-rewriting system $T_m(P)$, and hence, it is decidable by Theorem 3.11 whether this sequence has a solution. Observe that, if $(g_1, h_1), \ldots, (g_m, h_m)$ has a solution at all, then it has a solution φ that does not introduce any occurrences of the function symbol f by Lemma 5.5. Also the additional constant c does not cause any difficulties here as can easily be seen from the proofs of Theorems 3.9 and 3.11.

Assume next that $(g_1, h_1), \ldots, (g_m, h_m)$ contains a pair, say (g_1, h_1), such that $|g_1|_f > 0$ and $|h_1|_f > 0$, that is, $g_1 = w_1(f(g'_1, g''_1))$ and $h_1 = w_2(f(h'_1, h''_1))$ for some $w_1, w_2 \in \Sigma^*$. Then $(g_1, h_1), \ldots, (g_m, h_m)$ has a solution if and only if $w_1 \leftrightarrow^*_{T_m(P)} w_2$ holds, and if $(g_2, h_2), \ldots, (g_m, h_m), (g'_1, h'_1), (g''_1, h''_1)$ has a solution. If $w_1 \leftrightarrow^*_{T_m(P)} w_2$, then we are done, otherwise, we have to consider the sequence $(g_2, h_2), \ldots, (g_m, h_m), (g'_1, h'_1), (g''_1, h''_1)$ which is smaller than the original sequence $(g_1, h_1), \ldots, (g_m, h_m)$ with respect to the ordering $>$.

Finally, assume that $|g_i|_f = 0$ or $|h_i|_f = 0$ for each index $i = 1, \ldots, m$, and that there is a pair, say (g_1, h_1) such that $|g_1|_f > 0$. Then $g_1 = w_1(f(w_2, w_3))$ for some string $w_1 \in \Sigma^*$, and $h_1 = u_1(x)$ for a string $u_1 \in \Sigma^*$ and a variable $x \in X$ or $h_1 = u_1(c)$ for a string $u_1 \in \Sigma^*$. If $h_1 = u_1(c)$, then the pair (g_1, h_1) is obviously not unifiable mod $T_m(P)$, and hence, the whole sequence does not have a solution. So we may assume that $h_1 = u_1(x)$ for some variable $x \in X$. If this variable does occur in g_1, then the pair (g_1, h_1) is not unifiable mod $T_m(P)$, either, because of Lemma 5.6. So assume that $x \notin \mathrm{Var}(g_1)$. Since $T_m(P)$ is monadic and confluent, we can decide whether there exists some string $u_2 \in \Sigma^*$ such that $w_1 \leftrightarrow^*_{T_m(P)} u_1 u_2$. If no such string exists, then again by Lemma 5.6, (g_1, h_1) is not unifiable mod $T_m(P)$.

So assume that such a string u_2 does exist. We may assume without loss of generality that the strings w_1 and u_1 are irreducible mod $T_m(P)$. From Lemma 4.3 we obtain a finite set of irreducible strings $\mathrm{RM}(u_1)$ such that, whenever $u_1 u_2 \rightarrow^*_{T_m(P)} w_1$, then $u_1 = p_1 p'_1$, $u_2 = p'_2 p_2$ for some $p'_2 \in \mathrm{RM}(u_1)$, and $w_1 = p_1 d p_2$ for some $d \in \Sigma \cup \{\lambda\}$. Since w_1 has only finitely many suffixes, we conclude that there are only finitely many irreducible strings u_2 satisfying $u_1 u_2 \rightarrow^*_{T_m(P)} w_1$, and these strings can be determined effectively from u_1 and w_1. Let $\mathrm{Mul}(w_1, u_1)$ denote the set consisting of these strings.

Let ψ be a unifier for $(g_1, h_1) \bmod T_m(P)$, that is, $\psi(g_1) = w_1(f(\psi(w_2), \psi(w_3))) \Leftrightarrow^* u_1(\psi(x)) = \psi(h_1)$. Then $\psi(x) = u_2(f(h'_1, h''_1))$ for some string u_2, where $w_1 \leftrightarrow^*_{T_m(P)} u_1 u_2$, $\psi(w_2) \Leftrightarrow^* h'_1$, and $\psi(w_3) \Leftrightarrow^* h''_1$. Thus, u_2 belongs to the finite set $\mathrm{Mul}(w_1, u_1)$.

If, in addition, ψ is a solution for the sequence $(g_1, h_1), \ldots, (g_m, h_m)$, then ψ is also a solution for the sequence $(g'_2, h'_2), \ldots, (g'_m, h'_m)$, where g'_i (h'_i) is obtained from g_i (h_i) by replacing each occurrence of the variable x by the term $u_2(f(w_2, w_3))$. This sequence is smaller than the original sequence $(g_1, h_1), \ldots, (g_m, h_m)$ with respect to the ordering $>$, since

it does not contain any occurrences of the variable x, and it contains no variables that do not occur in the sequence $(g_1, h_1), \ldots, (g_m, h_m)$, either. Thus, the question of whether the original sequence has a solution is equivalent to the question whether one of the resulting sequences has a solution, where the variable x is replaced by a term $u_2(f(w_2, w_3))$ $(u_2 \in \mathrm{Mul}(w_1, u_1))$. This completes the proof of Theorem 5.7. □

Combining the results on the term-rewriting system $T_m(P)$ we obtain the following.

Corollary 5.8 *The finite, depth-reducing, and confluent term-rewriting system $T_m(P)$ has a decidable (simultaneous) E-unification problem, while the 2^{nd}-order E-matching problem for $T_m(P)$ is undecidable.*

Observe that the term-rewriting system $T_m(P)$ is linear and variable-preserving. To the best of our knowledge it yields the first known example of a non-collapsing theory with an undecidable 2^{nd}-order E-matching problem.

We can also interpret the system $T_m(P)$ as a term-rewriting system on the signature $\Sigma \cup \{c\}$, that is, we can delete the binary function symbol f and consider $T_m(P)$ as a string-rewriting system on the alphabet Σ. Theorem 3.11 and the proof of Theorem 5.4 show the following.

Corollary 5.9 *The finite, length-reducing, and confluent string-rewriting system $T_m(P)$ has a decidable simultaneous E-unification problem, while the simultaneous 2^{nd}-order E-matching problem for $T_m(P)$ is undecidable.*

Since $T_m(P)$ also has a decidable simultaneous E-matching problem (Theorem 3.9), it follows that Theorem 5.1 does not carry over to the simultaneous E-matching problem.

6 The 2^{nd}-order E-unification problem

Now we turn to the 2^{nd}-order E-unification problem for finite string-rewriting systems. For a string-rewriting system S on Σ, the 2^{nd}-order E-matching problem is reducible to the (1^{st}-order) E-matching problem in polynomial time, where for the latter S is considered as a string-rewriting system on $\Sigma \cup V$ (Theorem 5.1). Since the (1^{st}-order) E-matching problem is decidable in polynomial time for each finite, monadic, and confluent string-rewriting system, this means that the 2^{nd}-order E-matching problem is also decidable in polynomial time for each finite, monadic, and confluent string-rewriting system. However, this is not true for the 2^{nd}-order E-unification problem. This will be an immediate consequence of the following reducibility result.

Theorem 6.1 *The word matching problem for a string-rewriting system S on Σ is effectively reducible to the 2^{nd}-order E-unification problem for S, where Σ is extended by an additional free letter $\#$.*

Proof. Let $g \in (\Sigma \cup V)^*$ and $h \in \Sigma^*$ be an instance of the word matching problem for a string-rewriting system S on Σ. Let $\Gamma := \Sigma \cup \{\#\}$, where $\#$ is an additional letter, and let $c \in \Delta$ be a constant. We consider the instance $(g\#h\#g(c), h\#g\#h(c))$ of the 2^{nd}-order E-unification problem for S on $T_2(F_\Gamma, V, X)$.

Claim 1. If there is a morphism $\varphi : \{v \in V \mid |g|_v > 0\} \to \Sigma^*$ such that $\varphi(g) \leftrightarrow_S^* h$, then there also exists a 2^{nd}-order substitution ϕ satisfying $\phi(g\#h\#gc) \leftrightarrow_S^* \phi(h\#g\#hc)$.

Proof. For all $v \in V$, for which $|g|_v > 0$, define $\phi(v)$ through $\phi(v) := \varphi(v)(\Box)$. Then $\phi(g\#h\#gc) = \varphi(g)\#h\#\varphi(g)c \leftrightarrow_S^* h\#\varphi(g)\#hc = \phi(h\#g\#hc)$. □

Claim 2. If there is a 2^{nd}-order substitution ϕ satisfying $\phi(g\#h\#gc) \leftrightarrow_S^* \phi(h\#g\#hc)$, then there also exists a morphism $\varphi : \{v \in V \mid |g|_v > 0\} \to \Sigma^*$ such that $\varphi(g) \leftrightarrow_S^* h$.

Proof. Let ϕ be a 2^{nd}-order substitution satisfying $\phi(g\#h\#gc) \leftrightarrow_S^* \phi(h\#g\#hc)$. Assume first that there is a variable v such that $|g|_v > 0$ and $\phi(v) = wd$ for some $w \in (\Gamma \cup V)^*$ and $d \in (\Delta \cup X)$. Choose v in such a way that $g = g_1 v g_2$, and for all variables v' occurring in g_1, $\phi(v')$ ends in the place holder $\Box$. Then $\phi(g) = \phi(g_1)wd$, and so $\phi(g\#h\#gc) = \phi(g_1)wd$, while $\phi(h\#g\#hc) = h\#\phi(g_1)wd$. Thus, $|\phi(g\#h\#gc)|_\# < |\phi(h\#g\#hc)|_\#$. Since $\#$ is a free symbol for S, this contradicts the assumption that $\phi(g\#h\#gc) \leftrightarrow_S^* \phi(h\#g\#hc)$ holds. Thus, we see that $\phi(v) \in (\Gamma \cup V)^* \cdot \{\Box\}$ for all $v \in V$ occurring in g. Further, since $\#$ is a free symbol for S, $|\phi(v)|_\# = 0$ for all these variables $v \in V$. Also, if $|\phi(v)|_V > 0$ for some $v \in V$ occuring in g, then $|\phi(g)|_V > 0$. But $\phi(g\#h\#gc) = \phi(g)\#h\#\phi(g)c \leftrightarrow_S^* h\#\phi(g)\#hc = \phi(h\#g\#hc)$ implies that $\varphi(g) \leftrightarrow_S^* h$, where $\varphi : \{v \in V \mid |g|_v > 0\} \to (\Sigma \cup V)^*$ is defined through $\varphi(v) := w$ if $\phi(v) := w(\Box)$. However, $|h|_V = 0$, and hence, $\varphi(v) \in \Sigma^*$, that is, $\varphi : \{v \in V \mid |g|_v > 0\} \to \Sigma^*$ is a morphism satisfying $\varphi(g) \leftrightarrow_S^* h$. $\Box$

These two claims show that the given instance (g, h) of the word matching problem for S has a solution if and only if the instance $(g\#h\#gc, h\#g\#hc)$ of the 2^{nd}-order E-unification problem for S has a solution. This completes the proof of Theorem 6.1. $\Box$

In Section 4 we have seen that the finite, monadic, and confluent string-rewriting system $T_m(P)$ has an undecidable word matching problem. Thus, we obtain the following consequence.

Corollary 6.2 *The finite, monadic, and confluent string-rewriting system $T_m(P)$ has an undecidable 2^{nd}-order E-unification problem, when considered as a string-rewriting system on an alphabet containing at least one free symbol.*

In fact, using the recent result by Narendran and Otto [NaOt96] that there exists a finite, special, and confluent string-rewriting system with undecidable word matching problem, we even obtain the following stronger result.

Corollary 6.3 *There exists a finite, special, and confluent string-rewriting system for which the 2^{nd}-order E-unification problem is undecidable.*

Thus, the 2^{nd}-order E-unification problem is in general much more difficult than the 2^{nd}-order E-matching problem.

Next using an idea of Farmer [Far88] we show that the 2^{nd}-order E-unification problem for a string-rewriting system S reduces effectively to the word unification problem for S. Thus, the former cannot be more difficult than the latter.

A 2^{nd}-order substitution $\phi : V \cup X \to (\Sigma \cup V)^* \cdot (\Delta \cup \{\Box\} \cup X) \cup V$ is called **closed** if $\phi(v) \in \Sigma^* \cdot (\Delta \cup \{\Box\})$ for all $v \in V \cap \mathrm{dom}(\phi)$ and $\phi(x) \in \Sigma^* \cdot \Delta$ for all $x \in X \cap \mathrm{dom}(\phi)$. Recall that we assume that the set Δ of individual constants is non-empty. For a string $w \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$, $\mathrm{Var}_V(w) := \{v \in V \mid |w|_v > 0\}$, and $\mathrm{Var}_X(w) := \{x \in X \mid |w|_x > 0\}$.

Lemma 6.4 *Let S be a string-rewriting system on Σ, and let $g, h \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$. If there exists a 2^{nd}-order substitution ϕ satisfying $\phi(g) \leftrightarrow_S^* \phi(h)$, then there is also a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$, where $\mathrm{Var}_V(g)$, $\mathrm{Var}_V(h)$, $\mathrm{Var}_X(g)$ and $\mathrm{Var}_X(h)$ are contained in $\mathrm{dom}(\Psi)$.*

Proof. Let $g = g_1 c$ and $h = h_1 d$, where $g_1, h_1 \in (\Sigma \cup V)^*$ and $c, d \in \Delta \cup X$, and let ϕ be a 2^{nd}-order substitution unifying g and h modulo S. If $c \in \Delta$ and $\phi(g) = \phi(g_1 c) = \phi(g_1)c$, then let $a := c$; if $d \in \Delta$ and $\phi(h) = \phi(h_1 d) = \phi(h_1)d$, then let $a := d$, and otherwise, let

a denote some letter chosen from Δ. Observe that if the first two cases occur at the same time, then $\phi(g) \leftrightarrow_S^* \phi(h)$ implies that $c = d$, that is, the element a is uniquely defined by this definition. We now define another 2^{nd}-order substitution Ψ as follows, where $\mathrm{dom}(\Psi) := \mathrm{Var}_V(g) \cup \mathrm{Var}_V(h) \cup \mathrm{Var}_X(g) \cup \mathrm{Var}_X(h)$:

$$\text{if } c \in X, \text{ then we take } \Psi(c) := \begin{cases} \Pi_\Sigma(\phi(c)) \cdot a & \text{if } c \in \mathrm{dom}(\phi), \\ a & \text{if } c \notin \mathrm{dom}(\phi); \end{cases}$$

$$\text{if } d \in X, \text{ then we take } \Psi(d) := \begin{cases} \Pi_\Sigma(\phi(d)) \cdot a & \text{if } d \in \mathrm{dom}(\phi), \\ a & \text{if } d \notin \mathrm{dom}(\phi); \end{cases}$$

and for all $v \in \mathrm{dom}(\Psi) \cap V$, we take

$$\Psi(v) := \begin{cases} \Pi_\Sigma(\phi(v)) \cdot \Box & \text{if } v \in \mathrm{dom}(\phi) \text{ and } \phi(v) \in (\Sigma \cup V)^* \cdot \Box, \\ \Pi_\Sigma(\phi(v)) \cdot a & \text{if } v \in \mathrm{dom}(\phi) \text{ and } \phi(v) \in (\Sigma \cup V)^* \cdot (\Delta \cup X), \\ \Box & \text{if } v \notin \mathrm{dom}(\phi). \end{cases}$$

Here $\Pi_\Sigma : (\Sigma \cup V \cup \Delta \cup X \cup \{\Box\})^* \to \Sigma^*$ denotes the projection onto Σ^*.

Obviously, Ψ is a closed 2^{nd}-order substitution. It remains to verify that $\Psi(g) \leftrightarrow_S^* \Psi(h)$ holds. If $\phi(v) \in (\Sigma \cup V)^* \cdot \{\Box\}$ for all $v \in \mathrm{Var}_V(g) \cap \mathrm{dom}(\phi)$, then $\phi(g) = \phi(g_1) \cdot w \cdot e$, where $\phi(c) = w \cdot e$, $w \in (\Sigma \cup V)^*$ and $e \in (\Delta \cup X)$. Analogously, if $\phi(v) \in (\Sigma \cup V)^* \cdot \{\Box\}$ for all $v \in \mathrm{Var}_V(h) \cap \mathrm{dom}(\phi)$, then $\phi(h) = \phi(h_1) \cdot z \cdot f$, where $\phi(d) = z \cdot f$, $z \in (\Sigma \cup V)^*$ and $f \in (\Delta \cup X)$. Since $\phi(g) \leftrightarrow_S^* \phi(h)$, we can conclude that $e = f$. From the definition of Ψ we obtain $\Psi(g) = \Pi_\Sigma(\phi(g_1) \cdot w) \cdot a$ and $\Psi(h) = \Pi_\Sigma(\phi(h_1) \cdot z) \cdot a$. Since the symbols from V are interpreted as free symbols for S, we see that $\phi(g_1)we = \phi(g) \leftrightarrow_S^* \phi(h) = \phi(h_1)zf$ implies that $\Psi(g) = \Pi_\Sigma(\phi(g_1)w) \cdot a \leftrightarrow_S^* \Pi_\Sigma(\phi(h_1)z) \cdot a = \Psi(h)$ holds.

If $\phi(v) \notin (\Sigma \cup V)^* \cdot \{\Box\}$ for some $v \in \mathrm{Var}_V(h) \cap \mathrm{dom}(\phi)$, then let $h_1 = h_2 v h_3$ be chosen in such a way that $\phi(v) = v_1 f$ for some $v_1 \in (\Sigma \cup V)^*$ and $f \in (\Delta \cup X)$ and the prefix h_2 is of minimal length. Then $\phi(h) = \phi(h_2)v_1 f \leftrightarrow_S^* \phi(g) = \phi(g_1)we$, which again implies that $e = f$. Hence, $\Psi(g) = \Pi_\Sigma(\phi(g_1)w) \cdot a \leftrightarrow_S^* \Pi_\Sigma(\phi(h_2)v_1) \cdot a = \Psi(h)$.

If $\phi(v) \notin (\Sigma \cup V)^* \cdot \{\Box\}$ for some $v \in \mathrm{Var}_V(g) \cap \mathrm{dom}(\phi)$, the proof is completely analogous. $\Box$

Thus, for checking 2^{nd}-order E-unifiability modulo S, we can restrict our attention to closed 2^{nd}-order substitutions containing all the variables of the strings considered in their domain.

Theorem 6.5 *Let S be a finite string-rewriting system on Σ. Then the 2^{nd}-order E-unification problem for S reduces effectively to the word unification problem for S.*

Proof. Let $g = g_1 c$ and $h = h_1 d$ be an instance of the 2^{nd}-order E-unification problem for S, where $g_1, h_1 \in (\Sigma \cup V)^*$ and $c, d \in (\Delta \cup X)$. According to Lemma 6.4 there exists a 2^{nd}-order substitution ϕ satisfying $\phi(g) \leftrightarrow_S^* \phi(h)$ if and only if there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$, where $\mathrm{dom}(\Psi) = \mathrm{Var}_V(g) \cup \mathrm{Var}_V(h) \cup \mathrm{Var}_X(g) \cup \mathrm{Var}_X(h)$. Let V' denote the set of variables $v \in V$ that actually have an occurrence in g_1 or in h_1, and for each subset $U \subseteq V'$, let g_U and h_U denote the minimal prefix of g_1 and h_1, respectively, that contains an occurrence of a variable $v \in U$. Thus, if $\mathrm{Var}_V(g) \cap U \neq \emptyset$, then g_U is the shortest prefix of g_1 that ends in a variable $v \in U$, that is, $g_1 = g_2 v g_3$ for some $g_2 \in (\Sigma \cup (V' \setminus U))^*$ and $v \in U$, and if $\mathrm{Var}_V(g) \cap U = \emptyset$, then $g_U = g_1$, and similar for h_U. In particular, $g_\emptyset = g_1$ and $h_\emptyset = h_1$. Depending on c and d, we now distinguish between four cases.

Case 1: $c \in X$ and $d \in X$: For each subset $U \subseteq V'$, if $\mathrm{Var}_V(g) \cap U = \emptyset$, then replace $g_U = g_1$ by $g_U := g_1 c = g$, and if $\mathrm{Var}_V(h) \cap U = \emptyset$, then replace $h_U = h_1$ by $h_U := h_1 d = h$. Let

$I(g,h)$ denote the set of pairs $I(g,h) := \{(g_U, h_U) \mid U \subseteq V'\}$. This is a finite set that is easily obtained from g and h. Each of the pairs $(g_U, h_U) \in I(g,h)$ is now considered as an instance of the word unification problem for S, where the elements of $V' \cup \{c,d\}$ are interpreted as (string) variables.

Claim 1.1. If there is a pair $(g_U, h_U) \in I(g,h)$ such that the equation $g_U \sim h_U$ has a solution modulo S, then there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$.

Proof. Let $U \subseteq V'$, and let $\varphi : U' \to \Sigma^*$ be a morphism such that $\varphi(g_U) \leftrightarrow_S^* \varphi(h_U)$, where $U' := \{v \in V' \cup \{c,d\} \mid |g_U|_v + |h_U|_v > 0\}$. Since $c, d \in X$, we see that $g_U = g_2 v_1$ and $h_U = h_2 v_2$ for some variables $v_1, v_2 \in U'$, and $g = g_U g_3$ and $h = h_U h_3$ for some strings g_3 and h_3, respectively. We define a 2^{nd}-order substitution Ψ by taking

$$\Psi(v) := \begin{cases} \varphi(v) \cdot \square & \text{if } v \in U' \setminus \{v_1, v_2\}, \\ \varphi(v) \cdot a & \text{if } v \in \{v_1, v_2\}, \end{cases}$$

where a is a constant from Δ. Then Ψ is a closed 2^{nd}-order substitution, and $\Psi(g) = \Psi(g_2 v_1 g_3) = \varphi(g_2) \cdot \varphi(v_1) \cdot a = \varphi(g_U) \cdot a \leftrightarrow_S^* \varphi(h_U) \cdot a = \varphi(h_2) \cdot \varphi(v_2) \cdot a = \Psi(h_2 v_2 h_3) = \Psi(h)$. □

Claim 1.2. If there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$, then there is a pair $(g_U, h_U) \in I(g,h)$ such that the equation $g_U \sim h_U$ has a solution modulo S.

Proof. Let Ψ be a closed 2^{nd}-order substitution satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$. From the proof of Lemma 6.4 we see that we can assume without loss of generality that all the variables occurring in g and in h are contained in the domain of Ψ. Hence, $\Psi(g) = u_1 e$ and $\Psi(h) = u_2 e$ for some $u_1, u_2 \in \Sigma^*$ and $e \in \Delta$.

Let $U := \{v \in V' \mid \Psi(v) \in \Sigma^* \cdot \Delta\}$. Then $(g_U, h_U) \in I(g,h)$. We define a morphism $\varphi : V' \cup \{c,d\} \to \Sigma^*$ as follows:

$$\varphi(v) := \begin{cases} w & \text{if } v \in V', \text{ and } \Psi(v) = w \cdot \square, \\ w & \text{if } v \in U, \text{ and } \Psi(v) = w \cdot a \text{ for some } a \in \Delta, \\ w & \text{if } v \in \{c,d\}, \text{ and } \Psi(v) = w \cdot a \text{ for some } a \in \Delta. \end{cases}$$

Since $g_U = g_2 v_1$ and $h_U = h_2 v_2$ for some variables $v_1, v_2 \in U \cup \{c,d\}$, we see that $u_1 e = \Psi(g) = \Psi(g_2 v_1) = \Psi(g_2) \cdot \Psi(v_1)$ and $u_2 e = \Psi(h) = \Psi(h_2 v_2) = \Psi(h_2) \cdot \Psi(v_2)$. Since $e \in \Delta$ is a free constant, $\Psi(g) \leftrightarrow_S^* \Psi(h)$ implies that $\varphi(g_U) = \varphi(g_2 v_1) = \Psi(g_2) w_1 \leftrightarrow_S^* \Psi(h_2) w_2 = \varphi(h_2 v_2) = \varphi(h_U)$, where $\Psi(v_1) = w_1 \cdot e$ and $\Psi(v_2) = w_2 \cdot e$. □

Thus, there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$ if and only if at least one of the pairs $(g_U, h_U) \in I(g,h)$ is a positive instance of the word unification problem for S. This completes Case 1.

Case 2: One of c and d is a variable from X, while the other is a constant from Δ. By symmetry we may assume that $c \in X$ and $d \in \Delta$. For each subset $U \subseteq V'$, if $\text{Var}_V(g) \cap U = \emptyset$, then we replace $g_U = g_1$ by $g_U := g_1 c = g$. Let $I(g,h)$ denote the set of pairs $I(g,h) := \{(g_U, h_U) \mid U \subseteq V'\}$. Again this is a finite set that is easily obtained from g and h. Each of the pairs $(g_U, h_U) \in I(g,h)$ is now considered as an instance of the word unification problem for S, where the elements of $V' \cup \{c\}$ are interpreted as (string) variables.

Claim 2.1. If there is a pair $(g_U, h_U) \in I(g,h)$ such that the equation $g_U \sim h_U$ has a solution modulo S, then there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$.

Proof. Let $U \subseteq V'$, let $U' := \{v \in V' \cup \{c\} \mid |g_U|_v + |h_U|_v > 0\}$, and let $\varphi : U' \to \Sigma^*$ be a morphism such that $\varphi(g_U) \leftrightarrow_S^* \varphi(h_U)$. Then $g_U = g_2 v_1$ for some $v_1 \in U'$, and $h_U = h_2 v_2$ for

some $v_2 \in U$ or $h_U = h_1$, if $\mathrm{Var}_V(h) \cap U = \emptyset$. We define a 2^{nd}-order substitution Ψ by taking

$$\Psi(v) := \begin{cases} \varphi(v) \cdot \Box & \text{if } v \in U' \setminus \{v_1, v_2\}, \\ \varphi(v) \cdot d & \text{if } v \in \{v_1, v_2\}. \end{cases}$$

If $h_U = h_2 v_2$, then $\Psi(g) \leftrightarrow_S^* \Psi(h)$ follows as in the proof of Claim 1.1, and if $h_U = h_1$, then $\varphi(g_U) = \varphi(g_2 v_1) \leftrightarrow_S^* \varphi(h_1)$ implies that $\Psi(g) = \Psi(g_2 v_1) = \varphi(g_2 v_1) \cdot d \leftrightarrow_S^* \varphi(h_1) \cdot d = \Psi(h_1 d) = \Psi(h)$. □

Claim 2.2. If there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$, then there is a pair $(g_U, h_U) \in I(g, h)$ such that the equation $g_U \sim h_U$ has a solution modulo S.

Proof. Let $U := \{v \in V' \mid \Psi(v) \in \Sigma^* \cdot \Delta\}$. Then $(g_U, h_U) \in I(g, h)$, and it is easily checked that the following morphism $\varphi : V' \cup \{c\} \to \Sigma^*$ satisfies the congruence $\varphi(g_U) \leftrightarrow_S^* \varphi(h_U)$:

$$\varphi(v) := \begin{cases} w & \text{if } v \in V', \text{ and } \Psi(v) = w \cdot \Box, \\ w & \text{if } v \in U \cup \{c\}, \text{ and } \Psi(v) = w \cdot a \text{ for some } a \in \Delta. \end{cases}$$

□

This completes Case 2.

Case 3: $c, d \in \Delta$, and $c = d$. In this situation we let $I(g, h)$ denote the set $I(g, h) = \{(g_U, h_U) \mid U \subseteq V'\}$. As in Case 1 it can be shown that there exists a closed 2^{nd}-order substitution Ψ satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$ if and only if there is a pair $(g_U, h_U) \in I(g, h)$ such that the equation $g_U \sim h_U$ has a solution modulo S.

Case 4: $c, d \in \Delta$, but $c \neq d$. In this situation we let $I(g, h)$ denote the set $I(g, h) = \{(g_U, h_U) \mid U \subseteq V', U \neq \emptyset\}$. For each $(g_U, h_U) \in I(g, h)$, at least one of g_U and h_U ends with a variable $v \in U$. Observe that whenever Ψ is a closed 2^{nd}-order substitution satisfying $\Psi(g) \leftrightarrow_S^* \Psi(h)$, then $\Psi(v) \in \Sigma^* \cdot \Delta$ is satisfied for at least one variable $v \in V'$, since otherwise $\Psi(g) = w_1 c$ and $\Psi(h) = w_2 d$ could not be congruent modulo S. As in the other cases before, such a closed 2^{nd}-order substitution exists if and only if for some pair $(g_U, h_U) \in I(g, h)$, the equation $g_U \sim h_U$ has a solution modulo S.

Hence, in each of the four cases we have reduced the given instance of the 2^{nd}-order E-unification problem for S to a finite number of instances of the word unification problem for S. This completes the proof of Theorem 6.5. □

As we will see in the following the converse of Theorem 6.5 does not hold in general, that is, in general the word unification problem for a string-rewriting system S does not reduce to the 2^{nd}-order E-unification problem for S. We prove this result by presenting a particular example system S such that the 2^{nd}-order E-unification problem for S is decidable, while the word unification problem for S is undecidable.

Let $P = \{(y_i, z_i) \mid i = 2, \ldots, k\} \subseteq \{a, b\}^+ \times \{a, b\}^+$ be chosen in such a way that the modified Post Correspondence Problem MPCP(y_1, z_1) ($y_1, z_1 \in \{a, b\}^+$) is undecidable, let $\Sigma := \{a, b, e_2, \ldots, e_k, c, d, \$, \S, Z\}$, let $n := \max\{|y_i|, |z_i| \mid i = 2, \ldots, k\} + 1$, and let $T_\ell(P)$ denote the following string-rewriting system on Σ:

$$T_\ell(P) := \{e_i^n c \to c\rho(y_i), e_i^n d \to d\rho(z_i) \mid i = 2, \ldots, k\} \cup \{\$c \to \S, \$d \to \S\} \cup \{xZ \to Z \mid x \in \Sigma\}.$$

Here $\rho : \Sigma^* \to \Sigma^*$ denotes the function reversal. We claim that $T_\ell(P)$ has the following properties.

Theorem 6.6 *The string-rewriting system $T_\ell(P)$ defined above is finite, length-reducing, and confluent. The 2^{nd}-order E-unification problem for $T_\ell(P)$ is decidable, while the word unification problem for $T_\ell(P)$ is undecidable.*

Proof. It is easily checked that $T_\ell(P)$ is a finite length-reducing system that is confluent.

Claim 1. For all $y_1, z_1 \in \{a,b\}^+$ the following two statements are equivalent:

(a) MPCP(y_1, z_1) has a solution.

(b) There exists a string $w \in \Sigma^*$ such that $wc\rho(y_1) \leftrightarrow^*_{T_\ell(P)} wd\rho(z_1)$.

Proof. (a) $\Rightarrow$ (b): Let $i_1, \ldots, i_m \in \{2, \ldots, k\}$ such that $y_1 y_{i_1} \ldots y_{i_m} = z_1 z_{i_1} \ldots z_{i_m}$. Choose $w := \$e^n_{i_m} \ldots e^n_{i_1}$. Then $wc\rho(y_1) = \$e^n_{i_m} \ldots e^n_{i_1} c\rho(y_1) \rightarrow^m_{T_\ell(P)} \$c\rho(y_{i_m}) \ldots \rho(y_{i_1})\rho(y_1) \rightarrow_{T_\ell(P)} \S\rho(y_1 y_{i_1} \ldots y_{i_m}) = \S\rho(z_1 z_{i_1} \ldots z_{i_m}) \leftarrow_{T_\ell(P)} \$d\rho(z_{i_m}) \ldots \rho(z_{i_1})\rho(z_1) \leftarrow^m_{T_\ell(P)} \$e^n_{i_m} \ldots e^n_{i_1} d\rho(z_1) = wd\rho(z_1)$.

(b) $\Rightarrow$ (a): Let $w \in \Sigma^*$ satisfying $wc\rho(y_1) \leftrightarrow^*_{T_\ell(P)} wd\rho(z_1)$. We may assume without loss of generality that w is irreducible mod $T_\ell(P)$, and hence, either $w \in (\Sigma \setminus \{Z\})^*$ or $w = Zw_1$ for some $w_1 \in (\Sigma \setminus \{Z\})^*$. Since R is confluent, $wc\rho(y_1) \leftrightarrow^*_{T_\ell(P)} wd\rho(z_1)$ implies that $wc\rho(y_1) \rightarrow^*_{T_\ell(P)} u \leftarrow^*_{T_\ell(P)} wd\rho(z_1)$. If $w = Zw_1$, then we see from the form of the rules of $T_\ell(P)$, that $u = Zu_1$, where $w_1 c\rho(y_1) \rightarrow^*_{T_\ell(P)} u_1 \leftarrow^*_{T_\ell(P)} w_1 d\rho(z_1)$. It is now easily seen that w, respectively w_1, must end in $\$e^n_{i_m} \ldots e^n_{i_1}$ such that $y_1 y_{i_1} \ldots y_{i_m} = z_1 z_{i_1} \ldots z_{i_m}$. This completes the proof of Claim 1. □

From the choice of the set P and Claim 1 we immediately obtain the following undecidability result.

Claim 2. The word unification problem is undecidable for $T_\ell(P)$.

It remains to show that the 2^{nd}-order E-unification problem is decidable for $T_\ell(P)$. As a first step towards this goal we turn to the word matching problem for $T_\ell(P)$. So consider an existential sentence of the form

$$\exists v_1, \ldots, v_\ell : g_0 v_{i_1} g_1 \ldots v_{i_m} g_m \sim h,$$

where $g_0, g_1, \ldots, g_m, h \in \Sigma^*$ are irreducible, and $v_{i_1}, \ldots, v_{i_m} \in \{v_1, \ldots, v_\ell\}$. If $w_1, \ldots, w_\ell \in \Sigma^*$ is a solution for this sentence mod $T_\ell(P)$, then $g_0 w_{i_1} g_1 \ldots w_{i_m} g_m \rightarrow^*_{T_\ell(P)} h$. Now we distinguish between several cases.

(i) If $|h|_Z = 0$ and $|g_i|_Z > 0$ for some $i \in \{0, 1, \ldots, m\}$, then the existential sentence above cannot have a solution mod $T_\ell(P)$.

(ii) If $|h|_Z = 0$ and $|g_i|_Z = 0$ for all $i \in \{0, 1, \ldots, m\}$, then the existential sentence above has a solution $w_1, \ldots, w_\ell \in \Sigma^*$ if and only if it has a solution $w_1, \ldots, w_\ell \in (\Sigma \setminus \{Z\})^*$. Hence, the rules involving the letter Z are not used in the reduction $g_0 w_{i_1} g_1 \ldots w_{i_m} g_m \rightarrow^*_{T_\ell(P)} h$. Let Γ denote the subalphabet $\Gamma := \{a, b, \S\}$ of Σ. Then we see that $|g_0 w_{i_1} g_1 \ldots w_{i_m} g_m|_\Gamma \leq |h|_\Gamma$, and that each reduction step in the above reduction sequence strictly increases the Γ-length. Hence, there are only finitely many candidates for $w_1, \ldots, w_\ell$.

(iii) If $h = Zh_1$ and $j := \max\{i \mid |g_i|_Z > 0\}$, then the existential sentence above has a solution mod $T_\ell(P)$ if and only if one of the following existential sentences has a solution mod $T_\ell(P)$ that is Z-free:

$$g'_j v_{i_{j+1}} \ldots v_{i_m} g_m \sim h_1 \qquad (\text{where } g_j = Zg'_j),$$
$$v_{i_{j+k}} g_{j+k} \ldots v_{i_m} g_m \sim h_1 \quad (k \geq 1, v_{i_{j+k}} \notin \{v_{i_{j+k+1}}, \ldots, v_{i_m}\}).$$

(iv) If $h = Zh_1$ and $|g_i|_Z = 0$ for all $i \in \{0, 1, \ldots, m\}$, then the existential sentence above has a solution mod $T_\ell(P)$ if and only if one of the following existential sentences has a Z-free solution mod $T_\ell(P)$:

$$v_{i_j} g_j \ldots v_{i_m} g_m \sim h_1 \quad (j \geq 1,\ v_{i_j} \notin \{v_{i_{j+1}}, \ldots, v_{i_m}\}).$$

We see from case (ii) that cases (iii) and (iv) are solvable. Thus, we have the following decidability result.

Claim 3. The word matching problem is decidable for $T_\ell(P)$.

Finally, consider an instance $g, h \in (\Sigma \cup V)^* \cdot (\Delta \cup X)$ of the 2^{nd}-order E-unification problem for $T_\ell(P)$. If $|h|_X = 0$ and $|h|_V = 0$, then this is actually an instance of the 2^{nd}-order E-matching problem for $T_\ell(P)$, which is decidable by Claim 3 and by Theorem 5.1, since the 1^{st}-order E-matching problem is clearly a special case of the word matching problem. The same is true if $|g|_X = 0 = |g|_V$. Finally, if g as well as h both contain variables, then g and h are always 2^{nd}-order unifiable mod $T_\ell(P)$. Just take the 2^{nd}-order substitution ϕ that maps $v \mapsto Zc$ for each $v \in V$ occurring in g or in h and $x \mapsto Zc$ for each $x \in X$ occurring in g or in h, where $c \in \Delta \cup X$ is a fixed, but arbitrarily chosen individual constant or variable. Then $\phi(g)$ and $\phi(h)$ both end in the suffix Zc, and hence $\phi(g) \to^*_{T_\ell(P)} Zc \leftarrow^*_{T_\ell(P)} \phi(h)$. Thus, we have the following decidability result.

Claim 4. The 2^{nd}-order E-unification problem is decidable for $T_\ell(P)$.

This completes the proof of Theorem 6.6. □

Contrasting Theorem 6.5 and Theorem 6.6 we see that even for finite, length-reducing, and confluent string-rewriting systems, the word unification problem is strictly more difficult than the 2^{nd}-order E-unification problem.

However, we can at least establish a weak converse of Theorem 6.5, which says that from a finite string-rewriting system S on Σ, we can construct an extended system $S' := S \cup S_0$ on some extended alphabet $\Gamma \supsetneq \Sigma$ such that the word unification problem for S reduces to the 2^{nd}-order E-unification problem for S'. In the remaining part of this section we describe this construction and verify that it has the desired property.

Let S be a finite string-rewriting system on Σ, let Z and $\#$ be two additional symbols, and let Γ denote the alphabet $\Gamma := \Sigma \cup \{Z, \#\}$. Let S_0 denote the string-rewriting system $S_0 := \{Za \to Z, aZ \to Z \mid a \in \Sigma\}$, and let S' be the system $S' := S \cup S_0$. Then S' is a finite string-rewriting system on Γ that is easily obtained from S. Obviously, for all $u, v \in \Sigma^*$, $u \leftrightarrow^*_{S'} v$ if and only if $u \leftrightarrow^*_S v$, that is, restricted to Σ^*, S' is equivalent to S.

Theorem 6.7 *The word unification problem for S reduces effectively to the 2^{nd}-order E-unification problem for S'.*

Proof. Let $g, h \in (\Sigma \cup V)^*$ be an instance of the word unification problem for S. From g and h, we now construct an instance of the 2^{nd}-order E-unification problem for S' as follows.

Let $V' := \{v_1, \ldots, v_n\}$ be the set of variables that actually occur in g or in h. Define two 2^{nd}-order terms $y, z \in (\Gamma \cup V)^* \cdot (\Delta \cup X)$ as follows:

$$\begin{aligned} y &:= v_1 Z \# v_2 Z \# \ldots \# v_n Z \# v_1 Z \# \ldots \# v_n Z \# g \cdot a \text{ and} \\ z &:= Z v_1 \# Z v_2 \# \ldots \# Z v_n \# Z \# \ldots \# Z \# h \cdot a, \end{aligned}$$

where a is some constant from Δ, and where $|y|_\# = |z|_\# = 2n$. Obviously, the terms y and z are easily constructed from g and h.

Claim 1. If there is a morphism $\varphi : V' \to \Sigma^*$ satisfying $\varphi(g) \leftrightarrow_S^* \varphi(h)$, then there exists a 2^{nd}-order substitution ϕ such that $\phi(y) \leftrightarrow_{S'}^* \phi(z)$.

Proof. For $i = 1, \ldots, n$, let $\phi(v_i) := \varphi(v_i) \cdot \Box$. Then

$$\begin{array}{rcl} \phi(y) & = & \varphi(v_1)Z\#\ldots\#\varphi(v_n)Z\#\varphi(v_1)Z\#\ldots\#\varphi(v_n)Z\#\varphi(g)\cdot a \\ & \leftrightarrow_{S_0}^* & Z\#Z\#\ldots\#Z\#\varphi(g)\cdot a \\ & \leftrightarrow_S^* & Z\#Z\#\ldots\#Z\#\varphi(h)\cdot a \\ & \leftrightarrow_{S_0}^* & Z\varphi(v_1)\#\ldots\#Z\varphi(v_n)\#Z\#\ldots\#Z\#\varphi(h)\cdot a \\ & = & \phi(z). \end{array}$$

□

Claim 2. If there is a 2^{nd}-order substitution ϕ such that $\phi(y) \leftrightarrow_{S'}^* \phi(z)$, then there exists a morphism $\varphi : V' \to \Sigma^*$ satisfying $\varphi(g) \leftrightarrow_S^* \varphi(h)$.

Proof. Without loss of generality we can assume that ϕ is a closed 2^{nd}-order substitution such that $V' \subseteq \mathrm{dom}(\phi)$. Hence, $\phi(v_i) \in \Gamma^* \cdot (\Delta \cup \{\Box\})$ for all $i = 1, \ldots, n$.

Since $\#$ is a free function symbol for S', $\phi(y) \leftrightarrow_{S'}^* \phi(z)$ implies that $\phi(v_1 Z) \leftrightarrow_{S'}^* \phi(Zv_1)$. If $\phi(v_1) = w_1 \cdot b$ for some $w_1 \in \Gamma^*$ and $b \in \Delta$, then $\phi(v_1 Z) = w_1 \cdot b$ and $\phi(Zv_1) = Zw_1 \cdot b$. Hence, $|\phi(v_1 Z)|_Z < |\phi(Zv_1)|_Z$ contradicting the congruence $\phi(v_1 Z) \leftrightarrow_{S'}^* \phi(Zv_1)$. Thus, $\phi(v_1) = w_1 \cdot \Box$ for some $w_1 \in \Gamma^*$. Analogously, $\phi(v_i) = w_i \cdot \Box$ for some $w_i \in \Gamma^*$, $i = 2, \ldots, n$, which yields

$$\begin{array}{rcl} \phi(y) & = & w_1 Z\# w_2 Z\#\ldots\# w_n Z\# w_1 Z\# w_2 Z\#\ldots\# w_n Z\#\phi(ga) \text{ and} \\ \phi(z) & = & Zw_1\#Zw_2\#\ldots\#Zw_n\#Z\#\ldots\#Z\#\phi(ha). \end{array}$$

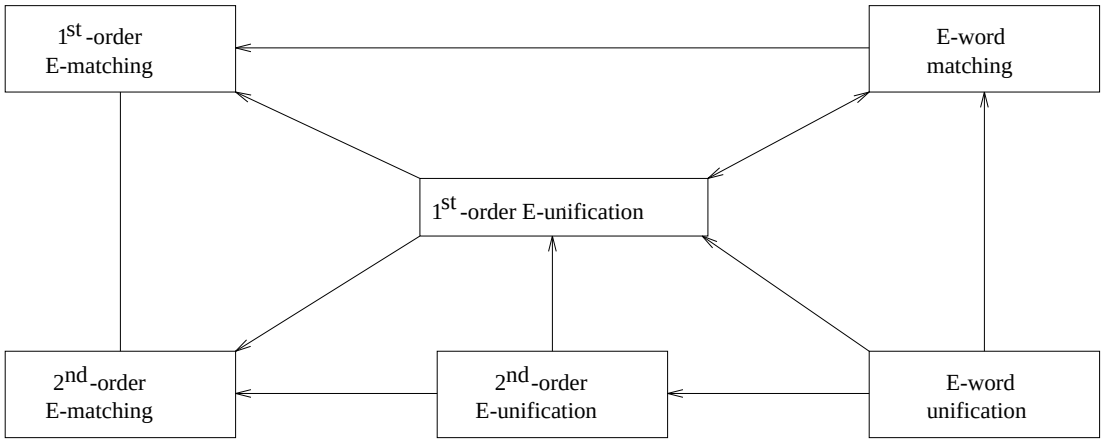
Without loss of generality we can assume that $w_1, \ldots, w_n$ are irreducible with respect to S_0. Since $\phi(y) \leftrightarrow_{S'}^* \phi(z)$ implies that $w_i Z \leftrightarrow_{S'}^* Zw_i$, $i = 1, \ldots, n$, and since the Z-length and the $\#$-length are not changed by applications of rules from S', we see that $\pi(w_i Z) = \pi(Zw_i)$, $i = 1, \ldots, n$, where $\pi : \Gamma^* \to \{Z, \#\}^*$ is the corresponding projection. Hence, $\pi(w_i Z) \in \{Z\}^+$, that is, $w_i = u_{i,0} Z u_{i,1} Z \ldots Z u_{i,n}$ for some $u_{i,j} \in \Sigma^*$, which in turn means that either $w_i = Z^{k_i}$ for some $k_i \geq 1$, or $w_i \in \Sigma^*$. However, since none of the strings w_i contains an occurrence of the symbol $\#$, $\phi(y) \leftrightarrow_{S'}^* \phi(z)$ also implies that $w_i Z \leftrightarrow_{S'}^* Z$, which in turn yields that $|w_i|_Z = 0$. Thus, for $i = 1, \ldots, n$, $\phi(v_i) = w_i \cdot \Box$ for some $w_i \in \Sigma^*$. This finally means that $\phi(ga) = \varphi(g) \cdot a$, where $\varphi : \{v_1, \ldots, v_n\} \to \Sigma^*$ is defined by taking $\varphi(v_i) := w_i$, $i = 1, \ldots, n$. Analogously, $\phi(ha) = \varphi(h) \cdot a$, and since a is a free constant, $\phi(y) \leftrightarrow_{S'}^* \phi(z)$ yields $\varphi(g) \leftrightarrow_{S'}^* \varphi(h)$, and hence, $\varphi(g) \leftrightarrow_S^* \varphi(h)$. □

Claims 1 and 2 show that the above construction is indeed a reduction from the word unification problem for S to the 2^{nd}-order E-unification problem for S'. □

If the string-rewriting system S is monadic, then so is the system S', and if S is noetherian and confluent, then again, so is S'. Thus, we see that for various classes of finite string-rewriting systems, the uniform versions of the word unification problem and the 2^{nd}-order E-unification problem are recursively equivalent. By uniform version we mean the decision problem that is obtained by providing the string-rewriting system as a part of the problem instance.

7 Conclusion

In the present paper we have considered various forms of equational matching and unification problems for string-rewriting systems, and we have compared them with respect to decidability. The results obtained can be summarized as shown in Diagram 1.



$\longleftarrow$ means "strictly weaker"
$\longrightarrow$ means "equivalent"
$\longleftrightarrow$ means "incomparable"
} with respect to recursive reductions

Diagram 1

In addition, we have presented some new decidability results for the (1^{st}-order) simultaneous E-matching and E-unification problems for finite, monadic, confluent string-rewriting systems in Section 3. In particular, we have considered 2^{nd}-order E-matching and E-unification problems for string-rewriting systems in Section 5 and Section 6.

We have seen that the 2^{nd}-order E-matching problem reduces to the 1^{st}-order E-matching problem. In contrast to this reducibility result the simultaneous version of the 2^{nd}-order E-matching problem for a string-rewriting system does not reduce to the simultaneous version of the 1^{st}-order E-matching problem (see Corollary 5.9).

Using a technique of Farmer [Far88] we have shown in Section 6 that the 2^{nd}-order E-unification problem for a string-rewriting system S reduces to the word unification problem for S. On the other hand, the word unification problem for S reduces to the 2^{nd}-order E-unification problem for an extended system $S' := S \cup S_0$ that is easily constructed from S. Thus, for various classes of finite string-rewriting systems, the uniform versions of the word unification problem and the 2^{nd}-order E-unification problem are recursively equivalent.

In all these investigations the 2^{nd}-order terms considered are built from the unary function constants that correspond to the letters of a finite alphabet Σ, some additional free individual constants Δ, the individual variables X, and the function variables V of arity 1. Thus, the 2^{nd}-order terms in $T_2(\Sigma \cup \Delta, V, X)$ are in 1-to-1 correspondence to the elements of the language $(\Sigma \cup V)^* \cdot (\Delta \cup X)$. One of the referees for [OND95] asked whether the results on 2^{nd}-order E-matching and 2^{nd}-order E-unification would remain the same, if function variables of arity larger than one were also taken into account. In the appendix we will answer this question in the affirmative.

For the 2^{nd}-order E-matching problem we prove that it still reduces to the 1^{st}-order E-matching problem, even if function variables of arity larger than one are used in forming 2^{nd}-order terms (Theorem 8.1). Then, using another idea of Farmer [Far88] it is shown that, for a string-rewriting system S, the 2^{nd}-order E-unification problem on the set of 2^{nd}-order terms $T_2(\Sigma \cup \Delta, \bigcup_{i \geq 1} V_i, X)$ reduces to finitely many instances of the same problem restricted to the set of 2^{nd}-order terms $T_2(\Sigma \cup \Delta, V_1, X)$. Here, for each $i \geq 1$, V_i denotes the set of function variables of arity i. Thus, the 2^{nd}-order E-matching problem and the 2^{nd}-order E-unification problem for string-rewriting systems do not get more difficult when function variables of arity larger than one are admitted.

References

[AvMa90] Avenhaus, J., and K. Madlener, "Term Rewriting and Equational Reasoning", in: R.B. Banerji (ed.), *Formal Techniques in Artificial Intelligence*, North-Holland, Amsterdam, 1990, pp. 1–43.

[BaSc96] Baader, F., and K. Schulz, "Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures," *Journal of Symbolic Computation* 21 (1996) 211–243.

[BaSi93] Baader, F., and J. Siekmann, "Unification Theory," in: D.M. Gabbay, C.J. Hogger and J.A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming,* Oxford University Press, Oxford, UK, 1993.

[BKN87] Benanav, D., D. Kapur, and P. Narendran, "Complexity of Matching Problems," *Journal of Symbolic Computation* 3 (1987) 203–216.

[Boo83] Book, R.V., "Decidable sentences of Church-Rosser congruences," *Theoretical Computer Science* 24 (1983) 301–312.

[BoOt93] Book, R.V., and F. Otto, *String-Rewriting Systems*, Springer, New York, 1993.

[Bür89] Bürckert, H.-J., "Matching–a special case of unification?" *Journal of Symbolic Computation* 8 (1989) 523-536.

[DeJo90] Dershowitz, N., and J.P. Jouannaud, "Rewrite Systems," in: J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics,* Elsevier, Amsterdam, 1990, pp.243–320.

[Fag87] Fages, F., "Associative-Commutative Unification," *Journal of Symbolic Computation* 3 (1987) 257–275.

[Far88] Farmer, W.M., "A unification algorithm for second-order monadic terms," *Annals of Pure and Applied Logic* 39 (1988) 131–174.

[For87] Fortenbacher, A., "An Algebraic Approach to Unification under Associativity and Commutativity," *Journal of Symbolic Computation* 3 (1987) 217–229.

[GaJo79] Garey, M.R., and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[Gol81] Goldfarb, W.D., "The undecidability of the second-order unification problem," *Theoretical Computer Science* 13 (1981) 225–230.

[HuOp80] Huet, G., and D. Oppen, "Equations and Rewrite Rules: a survey," in: R.V. Book (ed.), *Formal Languages: Perspectives and Open Problems*, Academic Press, New York, 1980, pp. 349–405.

[Mak77] Makanin, G.S., "The problem of solvability of equations in a free semigroup," *Math. USSR Sbornik* 32 (1977) 129–198.

[NaOt89] Narendran, P., and F. Otto, "Some polynomial-time algorithms for finite monadic Church-Rosser Thue systems," *Theoretical Computer Science* 68 (1989) 319–332.

[NaOt90] Narendran, P., and F. Otto, "Some results on equational unification," in: M. Stickel (ed.), *10th Int. Conf. on Automated Deduction*, Proceedings, Lecture Notes in Artificial Intelligence 449, Springer, Berlin, 1990, pp. 276–291. An extended version to appear in the *Journal of Automated Reasoning*.

[NaOt96] Narendran, P., and F. Otto, "The word matching problem for special and confluent string-rewriting systems is undecidable," in preparation.

[Ott86] Otto, F., "On two problems related to cancellativity," *Semigroup Forum* 33 (1986) 331–356.

[Ott95] Otto, F., "Solvability of word equations modulo finite special and confluent string-rewriting systems is undecidable in general," *Information Processing Letters* 53 (1995) 237–242.

[OND95] Otto, F., P. Narendran, and D.J. Dougherty, "Some independence results for equational unification," in: J. Hsiang (ed.), *Rewriting Techniques and Applications*, Proceedings, Lecture Notes in Computer Science 914, Springer, Berlin, 1995, pp. 367–381.

[PaWe78] Paterson, M., and M. Wegman, "Linear Unification," *Journal Computer System Sciences* 16 (1978) 158–167.

[Pec81] Pécuchet, J.-P., *Equations avec Constantes et Algorithme de Makanin*, Thèse 3e Cycle, Université de Rouen, France, 1981.

[Rob65] Robinson, J.A., "A Machine Oriented Logic Based on the Resolution Principle," *Journal of the Association for Computing Machinery* 12 (1965) 23–41.

[Sch90] Schulz, K.U., "Makanin's Algorithm for Word Equations – Two Improvements and a Generalization," in: K.U. Schulz (ed.), *Word Equations and Related Topics*, Proceedings, Lecture Notes in Computer Science 572, Springer, Berlin, 1990, pp. 85–150.

[Sie90] Siekmann, J.H., "An Introduction to Unification Theory," in: R.B. Banerji (ed.), *Formal Techniques in Artificial Intelligence*, North-Holland, Amsterdam, 1990, pp. 369–424.

[Sti81] Stickel, M.E., "A Unification Algorithm for Associative-Commutative Functions," *Journal of the ACM* 28 (1981) 423-434.

8 Appendix

In Sections 5 and 6 we have considered the 2^{nd}-order E-matching and E-unification problems for string-rewriting systems, where we only admitted function variables of arity 1. Here we discuss the situation of also having function variables of arity larger than 1. As we will see the results of Sections 5 and 6 essentially carry over to this more general case. This justifies the restriction to unary function variables adopted in the main body of the paper.

Let S be a finite string-rewriting system on some finite alphabet Σ. In order to discuss 2^{nd}-order terms we extend this alphabet by adding a non-empty set Δ of individual constants, a countably infinite set X of individual variables, and for each $n \geq 1$, a countably infinite set V_n of function variables of arity n. By interpreting each letter $a \in \Sigma$ as a unary function constant, we can form the set of 2^{nd}-order terms $T_2(\Sigma \cup \Delta, V, X)$, where V denotes the union

$V := \bigcup_{n \geq 1} V_n$. Thus, the 2^{nd}-order terms correspond to trees the leaves of which are labelled with individual constants or individual variables, and the inner nodes of which are labelled with function constants or function variables. Observe that the only nodes that have two or more sons are labelled with function variables.

With respect to the string-rewriting system S, the constants in Δ are treated as free constants, and the function variables in V are treated as free function symbols. Thus, S induces a reduction relation $\Longrightarrow_S^*$ on $T_2(\Sigma \cup \Delta, V, X)$, which is the reflexive and transitive closure of the following single-step reduction relation $\Longrightarrow_S$:

$$s \Longrightarrow_S t \text{ iff } \exists p \in O(s), \exists (\ell \to r) \in S, \exists u \in T_2(\Sigma \cup \Delta, V, X):$$

$$s|_p = \ell(u) \text{ and } t = s[r(u)]_p.$$

By $=_S$ we denote the congruence relation $\Longleftrightarrow_S^*$ on $T_2(\Sigma \cup \Delta, V, X)$ that is generated by the relation $\Longrightarrow_S$.

In order to discuss the notion of 2^{nd}-order substitutions we have to extend the 2^{nd}-order terms appropriately. Let $W := \{W_i \mid i \geq 1\}$ be a set of additional individual variables. Then $T_2(\Sigma \cup \Delta, V, X \cup W)$ denotes the set of **extended 2^{nd}-order terms**. For $t \in T_2(\Sigma \cup \Delta, V, X \cup W)$, the **rank** of t is the largest index m such that W_m actually occurs in t. Observe that the rank of t is zero if and only if $t \in T_2(\Sigma \cup \Delta, V, X)$.

A **2^{nd}-order substitution** is a mapping $\phi : V \cup X \to T_2(\Sigma \cup \Delta, V, X \cup W) \cup V$ satisfying the following conditions:

(1) $\text{dom}(\phi) := \{x \in V \cup X \mid \phi(x) \neq x\}$ is finite,

(2) $\phi(x) \in T_2(\Sigma \cup \Delta, V, X)$ for all $x \in X$, and

(3) $\phi(v)$ is a term of rank at most n for all $v \in V_n \cap \text{dom}(\phi)$.

The substitution ϕ can be extended to a mapping $\phi_e : T_2(\Sigma \cup \Delta, V, X) \to T_2(\Sigma \cup \Delta, V, X)$ as follows:

- if $g \in \Delta$, then $\phi_e(g) = g$,
- if $g \in X$, then $\phi_e(g) = \phi(g)$,
- if $g = ag_1$ for some $a \in \Sigma$, then $\phi_e(g) = a(\phi_e(g_1))$,
- if $g = v(g_1, \ldots, g_n)$ for some $v \in V_n$ such that $v \notin \text{dom}(\phi)$, then $\phi_e(g) = v(\phi_e(g_1), \ldots, \phi_e(g_n))$, and
- if $g = v(g_1, \ldots, g_n)$ for some $v \in V_n \cap \text{dom}(\phi)$, then $\phi_e(g) = \phi(v)[W_1 \leftarrow \phi_e(g_1), \ldots, W_n \leftarrow \phi_e(g_n)]$, that is, each occurrence of (the place holder) W_i in the term $\phi(v)$ is replaced by the term $\phi_e(g_i)$, $i = 1, \ldots, n$.

To simplify the notation the extension ϕ_e will simply be denoted by ϕ.

We want to establish the fact that the 2^{nd}-order E-matching problem and the 2^{nd}-order E-unification problem for S on $T_2(\Sigma \cup \Delta, V, X)$ are recursively reducible to the 2^{nd}-order E-matching problem, respectively the 2^{nd}-order E-unification problem, for S on $T_2(\Sigma \cup \Delta, V_1, X)$, that is, the presence of the function variables of arity larger than one does not make these decision problems more difficult. We begin by considering the matching problem.

Theorem 8.1 *The 2^{nd}-order E-matching problem for a string-rewriting system S on Σ is effectively reducible to the 1^{st}-order E-matching problem for S, where S is considered as a string-rewriting system on $\Sigma \cup V_1$.*

Proof. Let $g, h \in T_2(\Sigma \cup \Delta, V, X)$ constitute an instance of the 2^{nd}-order E-matching problem for S. First we consider the case that $\mathrm{Var}_V(h) = \emptyset$, that is, $h = h_1 d$ for some $h_1 \in \Sigma^*$ and $d \in X \cup \Delta$. If $\mathrm{Var}_V(g) = \emptyset$ as well, then $g = g_1 c$ for some $g_1 \in \Sigma^*$ and $c \in X \cup \Delta$. If $c \in \Delta$, then there exists a 2^{nd}-order substitution ϕ satisfying $\phi(g) \Longleftrightarrow_S^* h$ if and only if $c = d$ and $g_1 \longleftrightarrow_S^* h_1$. This is the word problem for S, which is reducible to the 1^{st}-order E-matching problem for S in the presence of the free letters in V_1 (cf. the proof of Theorem 5.1). If $c \in X$, then there exists a 2^{nd}-order substitution ϕ satisfying $\phi(g) \Longleftrightarrow_S^* h$ if and only if there exists a mapping $\varphi : \{c\} \to \Sigma^*$ such that $\varphi(g) = g_1 \cdot \varphi(c) \longleftrightarrow_S^* h_1$, that is, if and only if the existential sentence "$\exists v : g_1 v \sim h_1$" has a solution mod S. If $g = g_0 v(g_1, \ldots, g_n)$ for some $g_0 \in \Sigma^*$ and $v \in V_n$, then consider the existential sentence "$\exists v : g_0 v \sim h_1$". If there exists a string $w \in \Sigma^*$ such that $g_0 w \longleftrightarrow_S^* h_1$, then the 2^{nd}-order substitution $\phi : v \mapsto w(d)$ satisfies $\phi(g) = g_0 \phi(v) = g_0 w d \Longleftrightarrow_S^* h_1 d = h$. Conversely, if there exists a 2^{nd}-order substitution ϕ such that $\phi(g) \Longleftrightarrow_S^* h$, then we have $h = h_1 d \Longleftrightarrow_S^* \phi(g) = g_0 \phi(v(g_1, \ldots, g_n))$. Since $\mathrm{Var}_V(h) = \emptyset$, we see that $\phi(v(g_1, \ldots, g_n)) \in \Sigma^* \cdot \{d\}$, that is, $\phi(v(g_1, \ldots, g_n)) = wd$ for some $w \in \Sigma^*$. Thus, the substitution $\varphi : v \mapsto w$ gives a solution for the existential sentence above.

Finally, consider the case that $h = h_0 v(h_1, \ldots, h_n)$ for some $h_0 \in \Sigma^*$ and $v \in V_n$. If $\mathrm{Var}_V(g) = \mathrm{Var}_X(g) = \emptyset$, then obviously, there is no 2^{nd}-order match from g onto h. Otherwise, g can be written as $g = g_0 u(g_1, \ldots, g_m)$ for some $g_0 \in \Sigma^*$ and $u \in V_m$, or $g = g_0 x$ for some $g_0 \in \Sigma^*$ and $x \in X$.

Claim. There exists a 2^{nd}-order match from g onto h if and only if the existential sentence "$\exists v : g_0 v \sim h_0$" has a solution mod S.

Proof. If $w \in \Sigma^*$ satisfies $g_0 w \longleftrightarrow_S^* h_0$, then we take the 2^{nd}-order substitution ϕ defined by $u \mapsto wv(h_1, \ldots, h_n)$, respectively, $x \mapsto wv(h_1, \ldots, h_n)$. Then $\phi(g) = g_0 wv(h_1, \ldots, h_n) \Longleftrightarrow_S^* h_0 v(h_1, \ldots, h_n) = h$. Conversely, if ϕ is a 2^{nd}-order substitution satisfying $\phi(g) \Longleftrightarrow_S^* h$, then we see that $g_0 \phi(u(g_1, \ldots, g_m)) = \phi(g) \Longleftrightarrow_S^* h = h_0 v(h_1, \ldots, h_n)$. Hence, $\phi(u(g_1, \ldots, g_m)) = wv(w_1, \ldots, w_n)$ for some $w \in \Sigma^*$ and $w_1, \ldots, w_n \in T_2(\Sigma \cup \Delta, V, X)$ such that $g_0 w \longleftrightarrow_S^* h_0$, and $w_i \Longleftrightarrow_S^* h_i$, $i = 1, \ldots, n$. Thus, the mapping $v \mapsto w$ gives a solution for the existential sentence above. □

This completes the proof of Theorem 8.1. □

Comparing Theorem 8.1 to Theorem 5.1 we conclude that the 2^{nd}-order E-matching problem for string-rewriting systems does not get more difficult when function variables of arity larger than one are admitted.

Now we turn to the unification problem. Let (g, h) be an instance of the 2^{nd}-order E-unification problem for S. Let V' denote the set of function variables of arity larger than one that actually have occurrences in g or in h. To simplify the notation let $V' = \{v_1, \ldots, v_m\}$, where $v_i \in V_{j_i}$ $(j_i \geq 2,\ i = 1, \ldots, m)$. Let $v_1', \ldots, v_m' \in V_1$ be unary function variables that do not occur in g or in h, and let $K = K_1 \times \ldots \times K_m$, where $K_i = \{1, 2, \ldots, j_i\}$ $(i = 1, \ldots, m)$. For each m-tuple $\xi = (\xi_1, \ldots, \xi_m) \in K$ we define a 2^{nd}-order substitution σ_ξ with $\mathrm{dom}(\sigma_\xi) = V'$ as follows:

$$\sigma_\xi(v_i) := v_i'(W_{\xi_i}) \quad (i = 1, \ldots, m).$$

By J we denote the set of pairs of 2^{nd}-order terms $J := \{(\sigma_\xi(g), \sigma_\xi(h)) \mid \xi \in K\}$. Observe that, for each $\xi \in K$, $\sigma_\xi(g), \sigma_\xi(h) \in T_2(\Sigma \cup \Delta, V_1, X)$, that is, each pair $(\sigma_\xi(g), \sigma_\xi(h))$ can be interpreted as an instance of the 2^{nd}-order E-unification problem for S on the set of 2^{nd}-order terms $T_2(\Sigma \cup \Delta, V_1, X)$.

We will see that this set J gives the intended reduction from the instance (g, h) of the 2^{nd}-order E-unification problem for S on $T_2(\Sigma \cup \Delta, V, X)$ to finitely many instances of the 2^{nd}-order E-unification problem for S on $T_2(\Sigma \cup \Delta, V_1, X)$. The following two lemmata describe the correspondence between the solutions for (g, h) and those for the pairs in J.

Lemma 8.2 *If there exists an m-tuple $\xi \in K$ and a 2^{nd}-order substitution $\Psi : V_1 \cup X \to T_2(\Sigma \cup \Delta, V_1, X \cup \{W_1\}) \cup V_1$ such that $\Psi(\sigma_\xi(g)) =_S \Psi(\sigma_\xi(h))$, then there is a 2^{nd}-order substitution ϕ satisfying $\phi(g) =_S \phi(h)$.*

Proof. Let $\xi = (\xi_1, \ldots, \xi_m) \in K$, and let $\Psi : V_1 \cup X \to T_2(\Sigma \cup \Delta, V_1, X \cup \{W_1\}) \cup V_1$ be a 2^{nd}-order substitution satisfying $\Psi(\sigma_\xi(g)) =_S \Psi(\sigma_\xi(h))$. Define a 2^{nd}-order substitution ϕ through $\phi := \Psi \circ \sigma_\xi$, that is,

$$\phi(v) = \begin{cases} \Psi(v) & \text{if } v \in X \text{ or } v \in V_1 \cap (\mathrm{Var}_V(g) \cup \mathrm{Var}_V(h)), \\ \Psi(v_i')[W_1 \leftarrow W_{\xi_i}] & \text{if } v = v_i \in V', \end{cases}$$

where $\Psi(v_i')[W_1 \leftarrow W_{\xi_i}]$ denotes the term that is obtained from the term $\Psi(v_i')$ by replacing the variable W_1 by the variable W_{ξ_i}.

Claim. $\phi(g) =_S \Psi(\sigma_\xi(g))$.

Proof by induction on g:

- $g = c \in \Delta : \phi(g) = c = \Psi(\sigma_\xi(g))$.
- $g = x \in X : \phi(g) = \phi(x) = \Psi(x) = \Psi(\sigma_\xi(g))$.
- $g = a \cdot g_1$ for some $a \in \Sigma : \phi(g) = a \cdot \phi(g_1) =_S a \cdot \Psi(\sigma_\xi(g_1))$ (by the induction hypothesis) $= \Psi(\sigma_\xi(ag_1)) = \Psi(\sigma_\xi(g))$.
- $g = v \cdot g_1$ for some $v \in V_1 : \phi(g) = \phi(v \cdot g_1)$. If $\Psi(v) \in T_2(\Sigma \cup \Delta, V_1, X)$, then $\phi(v \cdot g_1) = \phi(v) = \Psi(v) = \Psi(v \cdot \sigma_\xi(g_1)) = \Psi(\sigma_\xi(v \cdot g_1))$. If $\Psi(v) = w \cdot W_1$, then $\phi(v \cdot g_1) = w \cdot \phi(g_1) =_S w \cdot \Psi(\sigma_\xi(g_1))$ (by the induction hypothesis) $= \Psi(v \cdot \sigma_\xi(g_1)) = \Psi(\sigma_\xi(v \cdot g_1))$. Finally, if $v \notin \mathrm{dom}(\Psi)$, then $v \notin \mathrm{dom}(\phi)$, either, and $\phi(v \cdot g_1) =_S \Psi(\sigma_\xi(v \cdot g_1))$ follows analogously.
- $g = v_i(g_1, \ldots, g_n)$, where $v_i \in V'$ and $n = j_i \geq 2$. Then $\sigma_\xi(g) = v_i' \sigma_\xi(g_{\xi_i})$, and hence, $\phi(g) = \Psi(v_i')[W_1 \leftarrow \phi(g_{\xi_i})] =_S \Psi(v_i')[W_1 \leftarrow \Psi(\sigma_\xi(g_{\xi_i}))]$ (by the induction hypothesis) $= \Psi(v_i' \cdot \sigma_\xi(g_{\xi_i})) = \Psi(\sigma_\xi(v_i(g_1, \ldots, g_n))) = \Psi(\sigma_\xi(g))$. □

Analogously, $\phi(h) =_S \Psi(\sigma_\xi(h))$. Thus, we see that $\phi(g) =_S \Psi(\sigma_\xi(g)) =_S \Psi(\sigma_\xi(h)) =_S \phi(h)$ holds, that is, ϕ is indeed a 2^{nd}-order substitution unifying the terms g and h mod S. □

It remains to establish the converse of Lemma 8.2.

Lemma 8.3 *If there is a 2^{nd}-order substitution ϕ satisfying $\phi(g) =_S \phi(h)$, then there exists an m-tuple $\xi \in K$ and a 2^{nd}-order substitution $\Psi : V_1 \cup X \to T_2(\Sigma \cup \Delta, V_1, X \cup \{W_1\}) \cup V_1$ such that $\Psi(\sigma_\xi(g)) =_S \Psi(\sigma_\xi(h))$ holds.*

Proof. Let ϕ be a 2^{nd}-order substitution such that $\phi(g) =_S \phi(h)$. We define an m-tuple $\xi := \xi(g, h) := (\xi_1, \ldots, \xi_m) \in K$ and a 2^{nd}-order substitution $\Psi : V_1 \cup X \to T_2(\Sigma \cup \Delta, V_1, X \cup \{W_1\}) \cup V_1$ inductively. First we choose a constant $c \in \Delta$:

$$c := \begin{cases} d & \text{if } \phi(g) = w_0 d \text{ for some } w_0 \in \Sigma^* \text{ and } d \in \Delta, \\ \text{some arbitrary element from } \Delta, & \text{otherwise.} \end{cases}$$

Observe that $\phi(g) = w_0 d$ implies that $\phi(h) = w_0' d$ for some $w_0' \in \Sigma^*$. We now construct $\xi(g) \in K$ and a 2^{nd}-order substitution Ψ_g inductively as follows:

- if $g = f \in \Delta$, then choose $\xi(g) := (1, \ldots, 1)$, and $\Psi_g := id$, the identity mapping;
- if $g = x \in X$, then choose $\xi(g) := (1, \ldots, 1)$, and $\Psi_g(x) := w_0 c$ if $\phi(x) = w_0 y$ or if $\phi(x) = w_0 v(w_1, \ldots, w_k)$ for some $w_0 \in \Sigma^*$ and $y \in \Delta \cup X$ or $v \in V_k$ $(k \geq 1)$;
- if $g = a g_1$ for some $a \in \Sigma$, then choose $\xi(g) := \xi(g_1)$, and $\Psi_g := \Psi_{g_1}$;
- if $g = v g_1$ for some $v \in V_1$, and $\phi(v) = w_0 W_1$ for some $w_0 \in \Sigma^*$, then choose $\xi(g) := \xi(g_1)$ and $\Psi_g(v) := \phi(v)$;
- if $g = v g_1$ for some $v \in V_1$, and $\phi(v) \notin \Sigma^* \cdot W_1$, then choose $\xi(g) := (1, \ldots, 1)$, and $\Psi_g(v) := w_0 c$, where $\phi(v) = w_0 d$ for some $w_0 \in \Sigma^*$ and $d \in \Delta \cup X$, or $\phi(v) = w_0 u(w_1, \ldots, w_k)$ for some $w_0 \in \Sigma^*$ and $u \in V_k$ $(k \geq 1)$;
- if $g = v_i(g_1, \ldots, g_k)$, where $v_i \in V' \cap V_k$, and $\phi(v_i) = w_0 W_j$ for some $w_0 \in \Sigma^*$ and $j \in \{1, \ldots, k\}$, then choose $\Psi_g(v_i') := w_0 W_1$ and $\xi(g) := \xi(g_j)|_{\xi_i = j}$, that is,

$$\xi(g) = (\xi(g_j)_1, \ldots, \xi(g_j)_{i-1}, j, \xi(g_j)_{i+1}, \ldots, \xi(g_j)_m);$$

- if $g = v_i(g_1, \ldots, g_k)$, where $v_i \in V' \cap V_k$, but $\phi(v_i) \notin \Sigma^* \cdot \{W_1, \ldots, W_k\}$, then choose $\xi(g) := (1, \ldots, 1)$, and $\Psi_g(v_i') := w_0 c$, where $\phi(v_i) = w_0 d$ for some $w_0 \in \Sigma^*$ and $d \in \Delta \cup X$, or $\phi(v_i) = w_0 u(w_1, \ldots, w_\ell)$ for some $w_0 \in \Sigma^*$ and $u \in V_\ell$ $(\ell \geq 1)$.

In this way we have constructed an m-tuple $\xi(g) \in K$ and a 2^{nd}-order substitution Ψ_g. By analyzing h in the same way this m-tuple $\xi(g)$ and this substitution Ψ_g are transformed into an m-tuple $\xi := \xi(g, h) \in K$ and a 2^{nd}-order substitution Ψ. Since ϕ is a 2^{nd}-order unifier mod S of g and h, the changes made to $\xi(g)$ and to Ψ_g in this second part of the construction do not introduce inconsistencies with respect to the first part of the construction. Certainly, $(\sigma_\xi(g), \sigma_\xi(h))$ is one of the instances of the 2^{nd}-order E-unification problem for S on the set of terms $T_2(\Sigma \cup \Delta, V_1, X)$ that are constructed from the pair (g, h). It remains to verify that $\Psi(\sigma_\xi(g)) =_S \Psi(\sigma_\xi(h))$ holds. We distinguish between two cases.

Claim 1. If $\phi(g) = w \cdot d$ for some $w \in \Sigma^*$ and $d \in \Delta$, then $\Psi(\sigma_\xi(g)) = \phi(g)$, and $\Psi(\sigma_\xi(h)) = w'd$ for some $w' \in \Sigma^*$ satisfying $w \longleftrightarrow_S^* w'$.

Proof. If $\phi(g) = w \cdot d$ for some $w \in \Sigma^*$ and $d \in \Delta$, then $\phi(g) =_S \phi(h)$ implies that $\phi(h) = w' \cdot d$ for some $w' \in \Sigma^*$ satisfying $w \longleftrightarrow_S^* w'$. Thus, it suffices to verify that $\Psi(\sigma_\xi(g)) = w \cdot c$ holds, since then the corresponding statement for h follows in the same way.

We proceed by induction on g:

- if $g = f \in \Delta$, then $\phi(g) = f = \Psi(\sigma_\xi(g))$;
- if $g = x \in X$, then $\phi(g) = \phi(x) = w \cdot d = w \cdot c = \Psi(x) = \Psi(\sigma_\xi(g))$;
- if $g = a g_1$ for some $a \in \Sigma$, then $\Psi(\sigma_\xi(g)) = a \cdot \Psi(\sigma_\xi(g_1)) = a w_1 d = a \cdot \phi(g_1) = \phi(a g_1)$ by the induction hypothesis;
- if $g = v g_1$ for some $v \in V_1$, then either $\phi(v) = w \cdot d$ or $\phi(v) = w_1 \cdot W_1$ for some prefix w_1 of w, that is, $w = w_1 w_2$ for some $w_2 \in \Sigma^*$, and $\phi(g_1) = w_2 d$. Then $\Psi(\sigma_\xi(g)) = \Psi(v \sigma_\xi(g_1)) = \left\{ \begin{array}{ll} w \cdot d & \text{if } \phi(v) = w \cdot d \\ w_1 \cdot \Psi(\sigma_\xi(g_1)) & \text{otherwise} \end{array} \right\} \underset{\text{I.H.}}{=} wd = \phi(g)$;
- if $g = v_i(g_1, \ldots, g_k)$ for some $v_i \in V' \cap V_k$, then either $\phi(v_i) = w \cdot d$ or $\phi(v_i) = w_1 \cdot W_j$, where $w = w_1 w_2$ and $\phi(g_j) = w_2 d$. In the former case $\Psi(\sigma_\xi(g)) = \Psi(v_i' \sigma_\xi(g_j)) = wd = \phi(g)$, and in the latter we have $\Psi(\sigma_\xi(g)) = \Psi(v_i' \sigma_\xi(g_j)) = w_1 \Psi(\sigma_\xi(g_j)) \underset{\text{I.H.}}{=} w_1 \phi(g_j) = w_1 w_2 d = \phi(g)$. □

Claim 2. If $\phi(g) = w \cdot x$ for some $w \in \Sigma^*$ and $x \in X$, then $\Psi(\sigma_\xi(g)) = w \cdot c$, and $\Psi(\sigma_\xi(h)) = w' \cdot c$ for some $w' \in \Sigma^*$ satisfying $w \longleftrightarrow_S^* w'$.

Proof. If $\phi(g) = w \cdot x$ for some $w \in \Sigma^*$, then $\phi(g) =_S \phi(h)$ implies that $\phi(h) = w' \cdot x$ for some $w' \in \Sigma^*$ satisfying $w \longleftrightarrow_S^* w'$. Thus, it suffices to verify that $\Psi(\sigma_\xi(g)) = w \cdot c$ holds, since then the corresponding statement for h follows analogously. However, this proof is simply done by induction on g as in the proof of Claim 1. □

Claim 3. If $\phi(g) = w_0\bar{v}(w_1, \ldots, w_k)$ for some $w_0 \in \Sigma^*$ and $\bar{v} \in V$, then $\Psi(\sigma_\xi(g)) = w_0 \cdot c$, and $\Psi(\sigma_\xi(h)) = w_0' \cdot c$ for some $w_0' \in \Sigma^*$ satisfying $w_0 \longleftrightarrow_S^* w_0'$.

Proof. $w_0\bar{v}(w_1, \ldots, w_k) = \phi(g) =_S \phi(h)$ implies that $\phi(h) = w_0'\bar{v}(w_1', \ldots, w_k')$ for some $w_0' \in \Sigma^*$ satisfying $w_0 \longleftrightarrow_S^* w_0'$, and hence, it suffices to verify that $\Psi(\sigma_\xi(g)) = w_0 \cdot c$ holds. We proceed by induction on g:

- if $g = d \in \Delta$, then $\phi(g) = d$ contradicting the hypothesis of Claim 3;
- if $g = x \in X$, then $\phi(g) = \phi(x) = w_0\bar{v}(w_1, \ldots, w_k)$ implies that $\Psi(\sigma_\xi(g)) = \Psi(x) = w_0 \cdot c$;
- if $g = ag_1$ for some $a \in \Sigma$, then $w_0\bar{v}(w_1, \ldots, w_k) = \phi(g) = \phi(ag_1) = a\phi(g_1)$ implies that $w_0 = a\bar{w}_0$ and $\phi(g_1) = \bar{w}_0\bar{v}(w_1, \ldots, w_k)$ hold. Hence, $\Psi(\sigma_\xi(g)) = a \cdot \Psi(\sigma_\xi(g_1)) = a \cdot \bar{w}_0 c$ (by the induction hypothesis) $= w_0 \cdot c$;
- if $g = vg_1$ for some $v \in V_1$, then either $\phi(v) = w_0\bar{v}(w_1, \ldots, w_k)$ implying that $\Psi(\sigma_\xi(g)) = \Psi(v\sigma_\xi(g_1)) = \Psi(v) = w_0 \cdot c$, or $\phi(v) = w_0'W_1$ for some prefix w_0' of w_0, that is, $w_0 = w_0'w_0''$ for some $w_0'' \in \Sigma^*$. In the latter case $w_0\bar{v}(w_1, \ldots, w_k) = \phi(g) = \phi(vg_1) = w_0'\phi(g_1)$ implies that $\phi(g_1) = w_0''\bar{v}(w_1, \ldots, w_k)$, and hence, we obtain $\Psi(\sigma_\xi(g_1)) = \Psi(v\sigma_\xi(g_1)) = w_0' \cdot \Psi(\sigma_\xi(g_1)) \underset{\text{I.H.}}{=} w_0'w_0'' \cdot c = w_0 \cdot c$;
- if $g = v_i(g_1, \ldots, g_\ell)$ for some $v_i \in V' \cap V_\ell$, then either $\phi(v_i) = w_0'W_j$ for some $j \in \{1, \ldots, \ell\}$ and $w_0' \in \Sigma^*$ satisfying $w_0 = w_0'w_0''$, or $\phi(v_i) = w_0\bar{v}(w_1', \ldots, w_k')$. In the former case $\xi(g)_i = j$ and $\Psi(\sigma_\xi(g)) = \Psi(v_i'\sigma_\xi(g_j)) = w_0'\Psi(\sigma_\xi(g_j)) \underset{\text{I.H.}}{=} w_0'w_0'' \cdot c = w_0 \cdot c$, since $w_0\bar{v}(w_1, \ldots, w_k) = \phi(v_i(g_1, \ldots, g_\ell)) = w_0'\phi(g_j)$ implies that $\phi(g_j) = w_0''\bar{v}(w_1, \ldots, w_k)$ holds. In the latter case $\Psi(\sigma_\xi(g)) = \Psi(v_i') = w_0 \cdot c$. □

Claims 1 to 3 yield $\Psi(\sigma_\xi(g)) =_S \Psi(\sigma_\xi(h))$. Actually, we have shown that Ψ can essentially be chosen to be a closed substitution, that is, for each variable $y \in (X \cup V) \cap \mathrm{dom}(\Psi), \mathrm{Var}_X(\Psi(y)) = \emptyset = \mathrm{Var}_V(\Psi(y))$. □

Lemma 8.2 and Lemma 8.3 together give the following result.

Theorem 8.4 *Let (g, h) be an instance of the 2nd-order E-unification problem for S. From (g, h) we can effectively determine a finite collection of instances $J = \{(g_\xi, h_\xi) \mid \xi \in K\}$ of the 2nd-order E-unification problem of S on the set of 2nd-order terms $T_2(\Sigma \cup \Delta, V_1, X)$ such that there exists a 2nd-order substitution ϕ satisfying $\phi(g) =_S \phi(h)$ if and only if, for some $\xi \in K$, there exists a (closed) 2nd-order substitution Ψ such that $\Psi(g_\xi) =_S \Psi(h_\xi)$.*

Thus, the 2nd-order E-unification problem for a string-rewriting system S does not get more complicated, if function variables of arity larger than one are added. Hence, as far as the decidability/undecidability of this problem is concerned, it suffices to admit function variables of arity one. This justifies the restriction placed on the 2nd-order terms in Sections 5 and 6.