Solving Trace Equations Using Lexicographical Normal Forms*

Volker Diekert

Universität Stuttgart,
Institut für Informatik
Breitwiesenstr. 20-22
70565 Stuttgart, Germany

Yuri Matiyasevich

Steklov Institute of Mathematics at
St. Petersburg, Fontanka 27,
St. Petersburg, 191011 Russia

Anca Muscholl

Universität Stuttgart,
Institut für Informatik
Breitwiesenstr. 20-22
70565 Stuttgart, Germany

Abstract

Very recently, Matiyasevich showed that the question whether an equation over a trace monoid has a solution or not is decidable. In his proof this question is reduced to the solvability of word equations with constraints, by induction on the size of the commutation relation. In the present paper we give another proof of this result using lexicographical normal forms. Our method is a direct reduction of a trace equation system to a word equation system with regular constraints. We use for this a new result on lexicographical normal forms.

*Report Nr. 1997/01, Fakultät Informatik, Universität Stuttgart.

1 Introduction

Solving equations is a central topic in various fields of computer science, especially concerning unification, as required by automated theorem proving or logic programming. An important and deep result has been obtained in 1977 by Makanin [10], who showed that the question whether an equation over words has a solution or not is decidable. I.e., there exists an algorithm deciding for a given equation $L = R$, where $L, R \in (\Omega \cup \Sigma)^*$ contain both unknowns from Ω and constants from Σ, whether an assignment $\sigma\colon \Omega \to \Sigma^*$ exists, satisfying $\sigma(L) = \sigma(R)$. Slightly more general, the existential theory of equations over free monoids is decidable, i.e., given an existentially quantified, closed first-order formula S over atomic predicates of the form $L = R$ and $L \neq R$, it is decidable whether S is valid over a given free monoid. Moreover, adding regular constraints, i.e., atomic predicates of the form $x \in C$, where C is a regular language, preserves decidability [13].
In this paper we prove the generalization of Makanin's result to trace monoids. Trace monoids have been originally studied in combinatorics [4] as quotients of free monoids by partial commutations. A free, partially commutative monoid (or trace monoid) is associated to an alphabet Σ and a symmetric, irreflexive commutation relation $I \subseteq \Sigma \times \Sigma$ as the quotient monoid $\mathbb{M}(\Sigma, I) = \Sigma^*/\{ab = ba \mid (a,b) \in I\}$. Trace monoids became meaningful for computer science in concurrency theory, where they were introduced by Mazurkiewicz [12] in connection with the semantic of labelled Petri nets. As a mathematical model traces are simple enough in order to be able to generalize many interesting results from free monoids in areas as automata theory or logic. However, traces are complex enough in order to require in most cases ingenious proof techniques and new ideas. For a broad overview of trace theory and related topics see "The Book of Traces" [6].
The results obtained so far in the area of equations on traces were restricted to equations without constants, i.e. of the form $L = R$ with $L, R \in \Omega^*$, and parametrized solution sets. A reason for this consists in the additional inherent difficulty introduced by partial commutations. An overview of some results obtained is presented in [5] (see also [7]). The decidability of the solvability of equations with constants was stated as an important open question. Very recently, Matiyasevich showed that the question whether an equation over a trace monoid has a solution or not is decidable [11]. In his proof this question is reduced to the solvability of word equations with constraints by eliminating partial commutativity. The original proof is an induction on the

size of the commutation relation I. In the present paper we give another proof of this result using a new result on lexicographical normal forms. Our method is a direct reduction of a trace equation system to a word equation system with regular constraints. The result on lexicographical normal forms is stated in the Main Lemma of Section 3. Section 4 contains the reduction from trace equations to word equations. Section 5 presents a detailed formula for concatenating lexicographical normal forms, leading to an estimation of the increase in the number of unknowns and equations. We conclude with two remarks concerning the parallel complexity of computing lexicographical normal forms.

2 Notations and Preliminaries

An *independence alphabet* is a pair (Σ, I), where Σ is a finite alphabet and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric relation, called *independence relation*. The complement $D = (\Sigma \times \Sigma) \setminus I$ is called *dependence relation*; it is a reflexive and symmetric relation.
With a given independence alphabet (Σ, I) we associate the *trace monoid* $\mathbb{M}(\Sigma, I)$. This is the quotient monoid $\Sigma^*/\equiv_I$, where $\equiv_I$ denotes the congruence being the equivalence relation generated by the set $\{uabv = ubav \mid (a,b) \in I,\ u,v \in \Sigma^*\}$; an element $t \in \mathbb{M}(\Sigma, I)$ is called a *trace*, the length $|t|$ of a trace t is given by the length of any representing word. By $\mathrm{alph}(t)$ we denote the alphabet of a trace t, being the set of letters occurring in t.
By 1 we denote both the empty word and the empty trace. Words $v, w \in \Sigma^*$ are called *independent* (w.r.t. I), if $\mathrm{alph}(v) \times \mathrm{alph}(w) \subseteq I$. In this case we simply write $(v,w) \in I$ or $v \in I(w)$ where $I(w)$ for $w \in \Sigma^*$ is a shorthand for $\{a \in \Sigma \mid (a,w) \in I\}$.
The *initial alphabet* of $w \in \Sigma^*$ is the set $\mathrm{init}(w) = \{a \in \Sigma \mid \exists w', w'' \in \Sigma^* \text{ with } w \equiv_I w' \text{ and } w' = aw''\}$.
A word language $L \subseteq \Sigma^*$ is called *I-closed* if whenever $v \in L$ and $w \equiv_I v$ then we have $w \in L$.

3 Lexicographical Normal Forms

Throughout the paper we will suppose that (Σ, I) denotes an independence alphabet, where Σ has the cardinality $n \geq 1$. We suppose that Σ is totally

ordered by $<$ and we identify Σ with the set $\{1, \ldots, n\}$. The order on Σ is extended to the lexicographical order on Σ^*.
A word $v \in \Sigma^*$ is in *lexicographical normal form* (w.r.t. I and $\leq$) if $v \leq w$ holds for all w such that $v \equiv_I w$. Let LNF denote the set of lexicographical normal forms, i.e., $\mathrm{LNF} \subseteq \Sigma^*$ is the set of minimal representatives for $\mathbb{M}(\Sigma, I)$. For $v \in \Sigma^*$ we denote by $\mathrm{Lnf}(v)$ the unique word $w \in \mathrm{LNF}$ such that $w \equiv_I v$. We view Lnf as a mapping $\mathrm{Lnf}\colon \Sigma^* \to \mathrm{LNF}$.
There is a simple characterization of lexicographical normal forms due to Anisimov and Knuth:

Proposition 3.1 ([3]) *Let Σ be totally ordered by $<$. Then a word $v \in \Sigma^*$ is in lexicographical normal form (w.r.t. I, $\leq$) if and only for every factor aub of v with $a, b \in \Sigma$, $u \in \Sigma^*$ and $(au, b) \in I$ we have $a < b$.*

Remark 3.2 Note that the above characterization yields a star-free expression for the set of lexicographical normal forms LNF (equivalently, a first-order formula defining the set LNF).

Definition 3.3 Let Σ be totally ordered by $<$.
For $\emptyset \neq A \subseteq \Sigma$ let the *height* $h(A)$ be $h(A) = \max\{a \mid a \in A\}$. Let also $h(\emptyset) = 0$. (Thus, $h(A) \in \{0, \ldots, n\}$.)
The *height* $h(v)$ of a word $v \in \Sigma^*$ is defined as $h(v) = h(\mathrm{alph}(v))$.

Our reduction of trace equations to word equations is based on the ability of concatenating lexicographical normal forms in a simple way. We give a formula for the product of lexicographical normal forms in the Main Lemma below and we will consider it in more details in Section 5. First, the following remark is clear:

Remark 3.4 Let $m \geq 1$ and $s, t, v, s_1, \ldots, s_m, t_1, \ldots, t_m \in \Sigma^*$ be words satisfying the following conditions:

$$\begin{aligned}
&s = s_1 \cdots s_m\,,\\
&t \equiv_I t_1 \cdots t_m\,,\\
&v = s_1 t_1 \cdots s_m t_m\,,\\
&t_j \in I(s_{j+1} \cdots s_m) \text{ for all } 1 \leq j < m\,.
\end{aligned}$$

Then we have $st \equiv_I v$.

The converse of the previous remark will be stated for lexicographical normal forms in the Main Lemma below. It is the crucial correctness argument for our reduction from trace equations to word equations. The important point is that the value of m can be bounded as a function in the size of the alphabet, and that the height decreases in the first $m-1$ factors.

Lemma 3.5 (Main Lemma) *Let $s, t, v \in \mathrm{LNF}$ be words in lexicographical normal form such that $st \equiv_I v$.*
Let $h = h(s)$ denote the height of s and suppose $h > 0$.
Then there exist an integer m, $1 \le m \le \frac{(n-1)(h-1)}{2} + 1$, and words $s_1, \ldots, s_m$, $t_1, \ldots, t_m \in \mathrm{LNF}$ in lexicographical normal form such that the following conditions hold:

$$\begin{aligned}
&s = s_1 \cdots s_m\,, \\
&t \equiv_I t_1 \cdots t_m\,, \\
&v = s_1 t_1 \cdots s_m t_m\,, \\
&s_i \neq 1, \ \textit{for all } \ 1 < i \le m\,, \\
&t_j \neq 1 \ \textit{for all } \ 1 \le j < m\,, \\
&t_j \in I(s_{j+1} \cdots s_m) \textit{ for all } \ 1 \le j < m\,, \\
&h(t_j) < h \textit{ for all } \ 1 \le j < m\,.
\end{aligned}$$

Remark 3.6 Before giving the proof of the Main Lemma, let us note that the trace equality $st \equiv_I v$ above cannot be replaced by word equalities of type $s = s_1 \cdots s_m$, $t = t_1 \cdots t_m$, $v = s_1 t_1 \cdots s_m t_m$. For example, consider $\mathbb{M}(\Sigma, I) = \{a, b, c\}^* / \{ab = ba, bc = cb\}$ and $s = c$, $t = ab$. Then the lexicographical normal form of st is $v = bca$.

Proof of the Main Lemma. We have $st \equiv_I v$ with $s, t, v \in \mathrm{LNF}$ and $h = h(s) > 0$. Consider the decomposition of v, $v = s_1 t_1 \cdots s_m t_m$, where $m \ge 1$ is minimal such that $s \equiv_I s_1 \cdots s_m$, $t \equiv_I t_1 \cdots t_m$, and $t_j \in I(s_{j+1} \cdots s_m)$ for all j, $1 \le j < m$. Clearly, since m is minimal, we have $s_i \neq 1$ and $t_j \neq 1$ for all $1 < i \le m$, $1 \le j < m$. Moreover, the words s_i, t_j are in lexicographical normal form.
Let us first show that $s = s_1 \cdots s_m$. Assume aub is a factor of $s_1 \cdots s_m$ with $a, b \in \Sigma$, $u \in \Sigma^*$ and $b \in I(au)$. If aub is a factor of some s_i, then $a < b$ follows by Prop. 3.1 and we are done. Otherwise let $i < j$ be such that $s_i \in \Sigma^* a u'$, $s_j \in u'' b \Sigma^*$ and $u = u' s_{i+1} \cdots s_{j-1} u''$. Since $t_k \in I(s_j)$ for $k < j$ we obtain $b \in I(au' s_{i+1} t_{i+1} \cdots s_{j-1} t_{j-1} u'')$, hence $a < b$ due to v being in lexicographical normal form. Thus $s_1 \cdots s_m$ is in lexicographical normal form, again by Prop. 3.1.

Suppose that $1 \leq j < m$ and let b denote the first letter of s_{j+1}. Let $a \in \mathrm{alph}(t_j)$, i.e. $t_j = uau'$ for some words u, u'. Then $au'b$ is a factor of $v \in \mathrm{LNF}$ satisfying $b \in I(au')$, thus we have $a < b$. Therefore $h(t_j) < h(b) \leq h$ for every $1 \leq j < m$.
Finally, assume by contradiction that $m > (n-1)(h-1)/2 + 1$. Let b_i, a_j denote the first letter of s_i, t_j respectively, $1 < i \leq m$, $1 \leq j < m$. Consider the chain of alphabets $I(s_2 \cdots s_m) \subseteq I(s_3 \cdots s_m) \subseteq \cdots \subseteq I(s_m)$. Note that we have $I(s_2 \cdots s_m) \neq \emptyset$ due to $t_1 \neq 1$, and also $I(s_m) \neq \Sigma$ due to $s_m \neq 1$. Therefore by the pigeon-hole principle there exist some indices i, j with $j - i \geq (h-1)/2$ satisfying $I(s_{i+1} \cdots s_m) = I(s_{j+1} \cdots s_m)$. Consider the factor $t_i s_{i+1} t_{i+1} \cdots t_j s_{j+1}$ of v. Note that $(t_k, s_l) \in I$ holds for every k, l such that $i \leq k, l-1 \leq j$, since $t_k \in I(s_{k+1} \cdots s_m) = I(s_{i+1} \cdots s_m)$. Therefore, $v \in \mathrm{LNF}$ implies $a_i < b_{i+1} < a_{i+1} < \cdots < a_j < b_{j+1}$ and we obtain $h(s) \geq h(b_{j+1}) \geq 2(j - i + 1) > h$, a contradiction. □

4 Trace Equation Systems

Definition 4.1 Let Σ denote a finite alphabet and Ω a finite set of unknowns, $\Sigma \cap \Omega = \emptyset$.

i) A *word equation over* Σ *and* Ω has the form $L = R$, with $L, R \in (\Sigma \cup \Omega)^*$.

ii) An *assignment for* an equation over Σ and Ω is a mapping $\sigma\colon \Omega \to \Sigma^*$ being extended in a natural way to a homomorphism $\sigma\colon (\Sigma \cup \Omega)^* \to \Sigma^*$, by $\sigma|_\Sigma = \mathrm{id}_\Sigma$.

   A *solution for* the equation $L = R$ is an assignment σ satisfying the equality $\sigma(L) = \sigma(R)$ in Σ^*.

Makanin [10] showed in 1977 that the question whether a word equation has a solution or not is decidable. Moreover, the solvability of a system of word equations can be reduced by well-known techniques to the solvability of a single equation. The problem can also be generalized by introducing regular constraints for the unknowns, i.e. regular sets $C_x \subseteq \Sigma^*$ for $x \in \Omega$. Here, a solution σ for an equation is required to satisfy $\sigma(x) \in C_x$ for all x. It has been shown by Schulz [13] that the solvability of word equations with regular constraints remains decidable. We are going to show that this more general result generalizes to traces.

Definition 4.2 Let (Σ, I) denote an independence alphabet and Ω a finite set of unknowns, $\Sigma \cap \Omega = \emptyset$.

i) A *trace equation over* (Σ, I) *and* Ω has the form $L \equiv R$, with $L, R \in (\Sigma \cup \Omega)^*$.

   A *solution for* the equation $L \equiv R$ is an assignment $\sigma\colon \Omega \to \Sigma^*$ satisfying $\sigma(L) \equiv_I \sigma(R)$.

ii) A *system of trace equations* is a formula built with the connectives **and** (&), **or** ($\vee$), **not** ($\neg$) over atomic predicates of the form $L \equiv R$ (trace equation) and $x \in C$ (constraint), where $C \subseteq \Sigma^*$ denotes an I-closed regular language.

   A *solution for* a system S over $(\Sigma, I), \Omega$ is an assignment $\sigma\colon \Omega \to \Sigma^*$ such that S evaluates to **true** when the atomic predicates $L \equiv R$, $x \in C$ are replaced by the truth value of $\sigma(L) \equiv_I \sigma(R)$, $\sigma(x) \in C$, respectively.

Remark 4.3 Later we will deal simultaneously with trace and word equations, so we distinguish notationally between $L = R$ for a word equation, whereas $L \equiv R$ denotes a trace equation. The difference is that equality under an assignment is interpreted in the free monoid Σ^*, resp. in the trace monoid $\mathbb{M}(\Sigma, I)$.

Remark 4.4 A system of word equations (with regular constraints) is just a special case of Def. 4.2 where one takes $I = \emptyset$. By Makanin's result (see the remarks after Def. 4.1), Schulz' generalization and the fact that negations can be eliminated, we note that the question whether a system of word equations has a solution or not is decidable.

Remark 4.5 Adding arbitrary regular constraints to a system of trace equations makes the question of solvability undecidable. This is due to the fact that the solvability of the equation $x \equiv y$ with $x \in C$, $y \in C'$ is equivalent to the non-emptiness of the intersection $\{w \in \Sigma^* \mid w \equiv_I v \text{ for some } v \in C\} \cap \{w \in \Sigma^* \mid w \equiv_I v \text{ for some } v \in C'\}$. For regular languages C, C' this last question is known to be undecidable, see [1].

Remark 4.6 Similar to the word case, the solvability of a trace equations system could be reduced to the solvability of a single trace equation (with additional constraints).

The aim of this section is to reduce the solvability problem for trace equations to word equations with regular constraints. We will give a direct proof using lexicographical normal forms to show the following

Theorem 4.7 ([11]) *Let S be a trace equation system over (Σ, I) and Ω. Then a set $\Omega' \supseteq \Omega$ of unknowns and a system of word equations S' over Σ, Ω' can be effectively constructed, such that S is solvable if and only if S' is solvable.*

Corollary 4.8 *The following problem is decidable:*

Instance: *An independence alphabet (Σ, I), a finite set of unknowns Ω, and a trace equation system S over $(\Sigma, I), \Omega$.*

Question: *Does S have a solution?*

4.1 Basic reductions

For a given trace equation system S we first eliminate constants by introducing new unknowns x_a and constraints $x_a \in \{a\}$, for $a \in \Sigma$. Then we replace a by x_a in each equation $L \equiv R$ of S. Clearly, for every solution σ of S the assignment $\sigma' = \sigma \cup \{x_a \mapsto a \mid a \in \Sigma\}$ is a solution for S'. Conversely, every solution σ' for S' yields a solution $\sigma = \sigma'|_\Omega$ for S.
Hence, without loss of generality atomic predicates are of the form

$$L \equiv R \quad \text{where } L, R \in \Omega^* .$$

Furthermore, we may assume that the given system is written in disjunctive normal form. Then we replace every negation $\mathbf{not}(L \equiv R)$ by the following formula

$$\begin{aligned} &\exists A, A', B, B' \subseteq \Sigma : \ (A \cap A' = \emptyset \ \& \ B \cup B' \neq \emptyset \ \& \\ &\quad L \equiv xy \ \& \ R \equiv xz \ \& \\ &\quad \mathrm{init}(y) = A \ \& \ \mathrm{init}(z) = A' \ \& \ \mathrm{alph}(y) = B \ \& \ \mathrm{alph}(z) = B') \quad (1) \end{aligned}$$

where x, y, z denote new unknowns. Clearly, constraints of the form $\mathrm{init}(x) = A$ or $\mathrm{alph}(x) = A$, $A \subseteq \Sigma$, can be expressed by I-closed regular languages.
Suppose that σ is a solution for S and $\sigma(L) \not\equiv_I \sigma(R)$. Let s denote the longest word such that for some $t, u \in \Sigma^*$:

$$\sigma(L) \equiv_I st \text{ and } \sigma(R) \equiv_I su .$$

Due to the maximality of s we obtain $\operatorname{init}(t) \cap \operatorname{init}(u) = \emptyset$. Moreover, $tu \neq 1$, since $st \not\equiv_I su$. Hence, $\sigma' = \sigma \cup \{x \mapsto s, y \mapsto t, z \mapsto u\}$ is a solution for the new subformula (1). Conversely, for every solution σ' of (1) one has $\sigma'(x)\sigma'(y) \not\equiv_I \sigma'(x)\sigma'(z)$, hence $\sigma'(L) \not\equiv_I \sigma'(R)$.
Since the set of I-closed regular languages forms an effective boolean algebra (as the family of recognizable subsets of a monoid [9]) we may also suppose that the formula contains no negated constraints, i.e. no formula of type **not**$(x \in C)$.
Moreover, it suffices to consider trace equations of the form $x_1 \cdots x_k \equiv y_1 \cdots y_l$ with $k \geq l > 0$, $x_i, y_j \in \Omega$. (The equation $x_1 \cdots x_k \equiv 1$ and the occurrences of x_i can be deleted from all equations, adding the constraints $\operatorname{alph}(x_i) = \emptyset$.)

4.2 From traces to words

The main idea for reducing trace equations to word equations will consist in replacing a trace equation $L \equiv R$ by some word equations $L_1 = R_1, \ldots, L_k = R_k$ with additional constraints and unknowns. Moreover, for every solution σ for $L \equiv R$ the mapping $\operatorname{Lnf} \circ \sigma \colon \Omega \to \Sigma^* \to \mathrm{LNF}$ can be extended to a solution for the equations $L_1 = R_1, \ldots, L_k = R_k$. Vice versa, each solution for the new equations will also be a solution for $L \equiv R$ when restricted to its unknowns.
This reduction actually goes by a chain of additional trace equations. By choosing an appropriate ordering we will show that the reduction process terminates yielding a system of word equations (with constraints).
We will consider in the following formulas $S(T, W, C)$ in disjunctive normal form with atomic predicates from some finite sets T, W, C, containing no negations. T will denote a set of trace equations, W a set of word equations and $C = \{x \in C_x \mid x \in \Omega\}$ a set of constraints, where each C_x is an I-closed regular language. Moreover, every $L \equiv R$ in T has the form $x_1 \cdots x_k \equiv y_1 \cdots y_l$ with $k \geq l \geq 1$, $x_i, y_j \in \Omega$. A solution for $S(T, W, C)$ is an assignment $\sigma \colon \Omega \to \Sigma^*$ which makes the formula evaluate to **true** when $(L \equiv R)$ from T, $(L = R)$ from W and $x \in C_x$ from C are replaced by the truth value of $\sigma(L) \equiv_I \sigma(R)$, $\sigma(L) = \sigma(R)$, and $\sigma(x) \in C_x$, respectively.

Definition 4.9 A formula $S(T, W, C)$ as above is called *normalized* if for every solution σ for S the mapping $\operatorname{Lnf} \circ \sigma$ is a solution for S, too.

Remark 4.10 Note that a formula $S(T, \emptyset, C)$ with I-closed constraints C is always normalized.

Remark 4.11 Suppose $S = S(T, W, C)$ is normalized and let $x \equiv y$ belong to T, where $x, y \in \Omega$. Consider the new formula $S' = S'(T', W', C)$ obtained from S by replacing every occurrence of $x \equiv y$ by $x = y$ and letting $T' = T \setminus \{x \equiv y\}$, $W' = W \cup \{x = y\}$. Then S is solvable if and only if S' is solvable. Note that a solution for S' is a solution for S, too. However, the converse is true only because S is a normalized system. Without this assumption about S it cannot be guaranteed that every solution for S also solves S'. Moreover, S' is a normalized system, too.

Example 4.12 Consider the trace equation system $S = (\{x \equiv y\}, \{x = ab, y = ba\}, \emptyset)$ given as the conjunction $(x \equiv y)$ & $(x = ab)$ & $(y = ba)$, where $(a, b) \in I$. Then S is not normalized, but of course it has a solution. However, replacing $x \equiv y$ by the word equation $x = y$ yields a system with no solution.

Proof of Thm. 4.7. Recall that an equation system with I-closed constraints $S = S(T, \emptyset, \{x \in C_x\}_{x \in \Omega})$ over $(\Sigma, I), \Omega$ is a normalized system. As previously noted it suffices to consider a formula S with trace equations of the form

$$x_1 \cdots x_k \equiv y_1 \cdots y_l, \quad k \geq l \geq 1, \quad (k, l) \neq (1, 1)\,. \qquad (2)$$

We suppose without loss of generality that for all unknowns $x \in \Omega$ some $A_x \subseteq \Sigma$ exists such that $h(A_x) > 0$, and $x \in C_x$ implies $\mathrm{alph}(x) = A_x$, for all x. Then $h(x)$ will mean $h(A_x)$. (In fact, we could also assume that $\mathrm{alph}(x) \subseteq A_x$. The obvious modifications needed below are left to the reader.) Moreover, let S be a conjunction of trace equations as in (2), of word equations and of I-closed regular constraints $x \in C_x$.
We define the *weight of* a trace equation $x_1 \cdots x_k \equiv y_1 \cdots y_l$ as in (2) as the triple of natural numbers $(l, h(x_1 \cdots x_{k-1}), k)$ and we consider the lexicographical ordering on $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$. We will show in the following that every such trace equation can be replaced by a formula over word equations and trace equations of lower weight, together with some additional constraints. Concretely, we apply the following rules.

Rule 1: Suppose $l > 1$ and let z denote a new unknown. Then we replace the equation $x_1 \cdots x_k \equiv y_1 \cdots y_l$ by

$$x_1 \cdots x_k \equiv z \quad \& \quad y_1 \cdots y_l \equiv z \quad \& \quad \mathrm{alph}(z) = \cup_{i=1}^{k} A_{x_i}\,.$$

Rule 2: Suppose $l = 1$ and $k > 2$, and let z denote a new unknown. Then we replace the equation $x_1 \cdots x_k \equiv y_1$ by

$$x_1 z \equiv y_1 \quad \& \quad x_2 \cdots x_k \equiv z \quad \& \quad \mathrm{alph}(z) = \cup_{i=2}^{k} A_{x_i} \,.$$

Rule 3: Suppose $l = 1$ and $k = 2$ and, in order to simplify notation, consider the equation $xy \equiv z$ (rather than uniformly $x_1 x_2 = y_1$). Moreover, let $h = h(x)$ denote the height of x.
We replace $xy \equiv z$ by the disjunction of the word equation

$$xy = z \tag{3}$$

and of formulas of the type

$$\begin{gathered} x = x_1 \cdots x_m \quad \& \quad y \equiv y_1 \cdots y_m \quad \& \quad z = x_1 y_1 \cdots x_m y_m \quad \& \\ \mathrm{alph}(x_1) = A_1 \quad \& \quad \cdots \quad \& \quad \mathrm{alph}(x_m) = A_m \quad \& \\ \mathrm{alph}(y_1) = B_1 \quad \& \quad \cdots \quad \& \quad \mathrm{alph}(y_m) = B_m \,, \end{gathered} \tag{4}$$

where x_i, y_j are new unknowns and the disjunction is taken over all values of m such that $1 < m \leq (n-1)(h-1)/2 + 1$ and over all alphabets $A_1, \ldots, A_m$, $B_1, \ldots, B_m \subseteq \Sigma$ such that

$$\begin{gathered} A_i \neq \emptyset \ \text{ for all } \ 1 < i \leq m, \ \text{ and} \\ 1 \leq h(B_j) < h \ \text{ for all } \ 1 \leq j < m, \ \text{ and} \\ B_j \times A_i \subseteq I \ \text{ for all } 1 \leq j < i \leq m, \ \text{ and} \\ A_1 \cup \cdots \cup A_m = \mathrm{alph}(x), \ \text{ and} \\ B_1 \cup \cdots \cup B_m = \mathrm{alph}(y) \,. \end{gathered} \tag{5}$$

The word equation $xy = z$ in (3) corresponds to the case $m = 1$ in (4) (this is in particular the case when $h = 1$ in (5)). It is actually the main case where the number of trace equations in S decreases.
Let S' denote the formula obtained from S by applying one of the three rules described above. Note that none of the rules adds negations.

Lemma 4.13 *Let S be a normalized equation system. Then the new system S' is normalized, too. Moreover, S' is solvable if and only if S is solvable.*

Proof. The claim is easily seen for the first two rules above, since there is a natural bijection between the set of solutions of S and of S', respectively.

Clearly, if S' has been obtained from S by the third rule, then every solution for S' is a solution for S, too, see Rem. 3.4. Therefore, let us consider an equation $xy \equiv z$ in S and a solution $\sigma\colon \Omega \to \Sigma^*$ for S. Then $\sigma' = \mathrm{Lnf} \circ \sigma$ is also solution for S, since S is normalized. We show that σ' can be extended to a solution for S'. Let $s = \sigma'(x)$, $t = \sigma'(y)$ and $v = \sigma'(z)$. Hence, $st \equiv_I v$ with $s, t, v \in$ LNF. If $h(s) = 1$, then in the Main Lemma we have $m = 1$, hence $v = st$. Therefore σ' is a solution of the new system S'.
Suppose that $st \equiv_I v$ with $s, t, v \in$ LNF, $h(s) = h > 1$. Then some m, $1 \leq m \leq (n-1)(h-1)/2+1$, and words $s_1, \ldots, s_m$, $t_1, \ldots, t_m$ exist, satisfying the conditions of the Main Lemma. With $\sigma'(x_i) = s_i$, $\sigma'(y_j) = t_j$ it is easily verified that σ' is a solution for S'.
The relation between the solution set of S and the solution set of S', together with the fact that S is normalized, imply that S' is normalized, too. This shows the lemma. $\square$

Finally, note that the new trace equation $y_1 \cdots y_m \equiv y$ in (4) has lower weight than $xy \equiv z$ due to $h(y_1 \cdots y_{m-1}) < h = h(x)$. Hence the reduction rules establish a noetherian rewriting system on trace equation systems. Applying the rules as long as possible we end with a system of word equations $S' = (\emptyset, W', C')$. $\square$

5 Computing Lexicographical Normal Forms

The aim of this section is to give a formula for computing the product of lexicographical normal forms. This yields an alternative proof of Thm. 4.7 and the so far best known upper bound on the number of new unknowns needed for the reduction. We conclude the section with two remarks concerning the parallel complexity of computing lexicographical normal forms.

Definition 5.1 Let $\sim_I$ be a relation on $(\Sigma^*)^*$ defined as

$$(x_1, \ldots, x_m) \sim_I (x'_1, \ldots, x'_{m'})$$

if $m = m'$ and there exists some i, $1 \leq i < m$ such that

$$\begin{aligned} x_j &= x'_j \quad \text{for all } 1 \leq j \leq m,\ j \notin \{i, i+1\}, \text{ and} \\ (x_i, x_{i+1}) &= (x'_{i+1}, x'_i) \text{ and } (x_i, x_{i+1}) \in I\,. \end{aligned}$$

By $\approx_I$ we denote the equivalence relation generated on $(\Sigma^*)^*$ by $\sim_I$.

Let $x \in \Sigma^*$, by abuse of language we write $(x_1, \ldots, x_m) \approx_I x$ if some words $x'_1, \ldots, x'_m$ exist such that

$$(x_1, \ldots, x_m) \approx_I (x'_1, \ldots, x'_m) \text{ and } x = x'_1 \cdots x'_m .$$

Theorem 5.2 *Let $s, t, v \in \mathrm{LNF}$ be words in lexicographical normal form such that $st \equiv_I v$.*
Then there exist positive integers m, p with $m \leq \frac{(n-1)^2}{2} + 1$, $p \leq n^n n!$ such that

$$\begin{aligned} &s = s_1 \cdots s_m , \\ &t = t_1 \cdots t_p , \\ &(s_1, \ldots, s_m, t_1, \ldots, t_p) \approx_I v , \end{aligned}$$

for some words $s_1, \ldots, s_m, t_1, \ldots, t_p \in \Sigma^*$.

Proof. Let $h = h(s)$ denote the height of s. Let $m(h), p(h)$ denote the minimal integers such that

$$\begin{aligned} &s = s_1 \cdots s_{m(h)} , \\ &t = t_1 \cdots t_{p(h)} , \\ &(s_1, \ldots, s_{m(h)}, t_1, \ldots, t_{p(h)}) \approx_I v , \end{aligned}$$

for some words s_i, t_j. Note that $m(h), p(h) \leq |v|$. For $h = 0$ we have $s = 1$, thus $m(0) = p(0) = 1$, which satisfies the theorem.
For $h \geq 1$ we will show by induction on h that $m(h) \leq (n-1)(h-1)/2 + 1$ and $p(h) \leq n^h h!$, thereby proving the theorem.
Let $h \geq 1$. By the Main Lemma there exist an integer $m \leq (n-1)(h-1)/2 + 1$ and words $s_1, \ldots, s_m$, $t_1, \ldots, t_m$ in lexicographical normal form satisfying

$$\begin{gathered} s = s_1 \cdots s_m , \\ t \equiv_I t_1 \cdots t_m , \\ v = s_1 t_1 \cdots s_m t_m , \\ s_i \neq 1,\ t_j \neq 1 \text{ for } 1 < i \leq m,\ 1 \leq j < m , \\ t_j \in I(s_{j+1} \cdots s_m) \text{ and } h(t_j) < h \text{ for } 1 \leq j < m . \end{gathered} \tag{6}$$

If $h = 1$, then $m = 1$ in (6), so we can take $m(h) = p(h) = 1$, since $t = t_1 \in \mathrm{LNF}$, which satisfies the claim. Hence let $h, m \geq 2$.
Let $\bar{t}_1 = t_1$ and $\bar{t}_i = \mathrm{Lnf}(\bar{t}_{i-1} t_i)$ for $i = 2, \ldots, m$. Clearly, $\bar{t}_m = t$, $h(\bar{t}_i) < h$ for $1 \leq i < m$ and

$$\bar{t}_{i-1} t_i \equiv_I \bar{t}_i, \quad \text{for } 1 < i \leq m. \tag{7}$$

Now we can apply the induction hypothesis to each of the $(m-1)$ equivalences (7) obtaining
$$t \approx_I (t'_1, \ldots, t'_p)\,, \tag{8}$$
for some $p \leq (m-1)[m(h-1)+p(h-1)]$, some words $t'_1, \ldots, t'_p$ and some integers $1 = l_0 < l_1 < \cdots < l_m = p+1$ such that

$$t_i = t'_{l_{i-1}} \cdots t'_{l_i - 1} \text{ for every } 1 \leq i \leq m\,. \tag{9}$$

The above claim can be verified by noting that

$$t \approx_I (t'_1, \ldots, t'_i, \ldots, t'_j, \ldots, t'_q) \text{ and } t'_i \cdots t'_j \approx_I (v_1, \ldots, v_k)$$

implies that
$$t \approx_I (t'_1, \ldots, t'_{i-1}, v'_1, \ldots, v'_l, t'_{j+1}, \ldots, t'_q)\,,$$
for some $l \leq j - i + k$ and $v'_1, \ldots, v'_l \in \Sigma^*$, such that $v'_1 \cdots v'_l = v_1 \cdots v_k$ and each v'_q is a factor of some v_r.
Hence, we obtain from (8), (9) for suitable words $t''_1, \ldots, t''_p$:

$$\begin{array}{rcl} t & = & t''_1 \cdots t''_p\,, \\ v & \approx_I & (s_1, \ldots, s_m, t_1, \ldots, t_m) \approx_I (s_1, \ldots, s_m, t'_1, \ldots, t'_p) \\ & \approx_I & (s_1, \ldots, s_m, t''_1, \ldots, t''_p)\,. \end{array}$$

Hence by the induction hypothesis we get

$$\begin{array}{rcl} p(h) & \leq & (m-1)[m(h-1)+p(h-1)] \\ & \leq & (n-1)(h-1)/2\ [(n-1)(h-2)/2 + 1 + n^{h-1}(h-1)!] \\ & \leq & n^h h!\,, \end{array}$$

which concludes the proof. □

Remark 5.3 We can also use Thm. 5.2 in order to prove the main result, Thm. 4.7. Recall that the main difficulty consists in replacing a trace equation of the form $xy \equiv z$, where $x, y, z \in \Omega$. By Thm. 5.2 we simply replace such an equation $xy \equiv z$ by a disjunction over clauses of the form

$$\begin{array}{l} x = x_1 \cdots x_m \quad \& \quad y = y_1 \cdots y_p \quad \& \\ z = z_{\pi(1)} \cdots z_{\pi(m+p)} \quad \& \quad \text{alph}(x_i) = A_i \quad \& \quad \text{alph}(y_j) = B_j\,, \end{array}$$

for all $1 \leq m \leq \frac{(n-1)^2}{2} + 1$, $1 \leq p \leq n^n n!$, $\pi \in S^I_{m+p}$ and $A_i, B_j \subseteq \Sigma$. Here x_i, y_j denote new variables and $z_i = x_i$ for $1 \leq i \leq m$, resp. $z_{m+j} = y_j$ for $1 \leq j \leq p$. S^I_{m+p} denotes the set of permutations over $\{1, \ldots, m+p\}$ such that for $i < j$ the inequality $\pi(i) > \pi(j)$ implies $(z_i, z_j) \in I$. This reduction of a single trace equation to word equations roughly yields an increase in the number of word equations by $(N+2)!2^{n(N+1)}$, where $N = n^n n! + (n-1)^2/2 + 1$. Hereby we need N additional unknowns.

We conclude this section with two remarks concerning the parallel complexity of computing lexicographical normal forms. We consider uniform circuit complexity classes like AC^0 and TC^0. Let $f\colon \Sigma^* \to \Sigma^*$ be a function such that $|f(w)| = p(|w|)$ for some polynomial p and every $w \in \Sigma^*$. Let $k \geq 0$. Then f is AC^k-computable if there is a family $(C_n)_{n\geq 0}$ of polynomial-size circuits of depth $O(\log^k(n))$ with AND and OR gates of unbounded fan-in/out and unary NOT gates, such that $C_{|w|}$ computes $f(w)$ for all $w \in \Sigma^*$. A function f is TC^k-computable if there is a family of circuits as above which in addition to AND, OR and NOT gates contain MAJORITY gates of unbounded fan-in/out. A MAJORITY gate yields 1 if and only if more than half of its inputs are 1. In order to be able to deal with arbitrary alphabets Σ one usually assumes that the circuits have special input/output gates testing $x = a$ for each input position x and letter $a \in \Sigma$ (analogously for the outputs). Uniformity means that given $n \geq 0$ (a fixed coding of) the circuit C_n can be easily computed (e.g. in logarithmic space). It is not very hard to verify that $\mathrm{AC}^k \subseteq \mathrm{TC}^k \subseteq \mathrm{AC}^{k+1}$, $k \geq 0$. For more details about circuit complexity see e.g. [14]. We state the results below without proofs, which will appear elsewhere. With Thm. 5.2 we obtain

Corollary 5.4 *Let (Σ, I) denote an independence alphabet.*
Then we can compute $\mathrm{Lnf}(st)$ on input $s, t \in \mathrm{LNF}$ in uniform AC^0.

Proof. Since m, p in Thm. 5.2 are bounded by the size of the alphabet we can build a subcircuit for each factorization $s = s_1 \cdots s_m$, $t = t_1 \cdots t_p$ and each permutation in S^I_{m+p}. Moreover, by Prop. 3.1 we can test whether a given word is in lexicographical normal form in AC^0, too. $\square$

We could apply Cor. 5.4 in order to compute the function Lnf in AC^1. However, we can do better: the mapping $\mathrm{Lnf}\colon \Sigma^* \to \mathrm{LNF}$ is computable in uniform TC^0. This result can be compared with the fact that the equivalence $s \equiv_I t$ can be verified in uniform TC^0, too (see [2]).

Proposition 5.5 *Let (Σ, I) denote an independence alphabet. Then we can compute* $\mathrm{Lnf}(s)$ *on input* $s \in \Sigma^*$ *in uniform* TC^0.

Proof. For $w \in \Sigma^+$ and $1 \leq i, j \leq |w|$ let $\mathrm{lex}(i, j)$ denote the predicate which is satisfied by w if and only if the ith position of w precedes the jth position in the lexicographical normal form of w, $\mathrm{Lnf}(w)$. Then $\mathrm{lex}(i, j)$ is first-order definable, see [8, Prop. 3]. But first-order definable predicates are AC^0-computable (see e.g. [14] for more details about the relation between circuit complexity and first-order logic). Finally, the ith position of a word w occurs as the jth position of $\mathrm{Lnf}(w)$ if and only if $j = \#\{k \mid \mathrm{lex}(k, i)\} + 1$. The last statement is easily seen to be checkable in TC^0. $\square$

References

[1] IJ. J. Aalbersberg and H. J. Hoogeboom. Characterizations of the decidability of some problems for regular trace languages. *Mathematical Systems Theory*, 22:1–19, 1989.

[2] C. Àlvarez and J. Gabarró. The parallel complexity of two problems on concurrency. *Information Processing Letters*, 38:61–70, 1991.

[3] A. V. Anisimov and D. E. Knuth. Inhomogeneous sorting. *International Journal of Computer and Information Sciences*, 8:255–260, 1979.

[4] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements.* Number 85 in Lecture Notes in Mathematics. Springer, Berlin-Heidelberg-New York, 1969.

[5] C. Choffrut. Combinatorics in trace monoids I. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, chapter 3, pages 71–82. World Scientific, Singapore, 1995.

[6] V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.

[7] C. Duboc. On some equations in free partially commutative monoids. *Theoretical Computer Science*, 46:159–174, 1986.

[8] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.

[9] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, New York and London, 1974.

[10] G. S. Makanin. The problem of solvability of equations in a free semigroup. *Math. Sbornik*, 103:147–236, 1977. English transl. in Math. USSR Sbornik 32 (1977).

[11] Yu. Matiyasevich. Reduction of trace equations to word equations, 1996. Talk given at the "Colloquium on Computability, Complexity, and Logic", 5./6. December 1996, Institut für Informatik, Universität Stuttgart, Germany.

[12] A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.

[13] K. U. Schulz. Makanin's algorithm for word equations — Two improvements and a generalization. In K. U. Schulz, editor, *Word Equations and Related Topics*, number 572 in Lecture Notes in Computer Science, pages 85–150, Berlin-Heidelberg-New York, 1991. Springer.

[14] H. Straubing. *Finite automata, formal logic, and circuit complexity.* Birkhäuser, 1994.