Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 332 (2005) 47–61
Theoretical Computer Science
www.elsevier.com/locate/tcs

Aspects of shuffle and deletion on trajectories

Lila Kari[a], Petr Sosík[a, b, *]

[a] *Department of Computer Science, The University of Western Ontario, London, ON N6A 5B7, Canada*
[b] *Institute of Computer Science, Silesian University, 74601 Opava, Czech Republic*

Received 15 March 2004; received in revised form 13 September 2004; accepted 23 September 2004
Communicated by A. Salomaa

Abstract

Word and language operations on trajectories provide a general framework for the study of properties of sequential insertion and deletion operations. A *trajectory* gives a syntactical constraint on the scattered insertion (deletion) of a word into(from) another one, with an intuitive geometrical interpretation. Moreover, deletion on trajectories is an inverse of the shuffle on trajectories. These operations are a natural generalization of many binary word operations like catenation, quotient, insertion, deletion, shuffle, etc. Besides they were shown to be useful, e.g. in concurrent processes modelling and recently in biocomputing area.

We begin with the study of algebraic properties of the deletion on trajectories. Then we focus on three standard decision problems concerning linear language equations with one variable, involving the above mentioned operations. We generalize previous results and obtain a sequence of new ones. Particularly, we characterize the class of binary word operations for which the validity of such a language equation is (un)decidable, for regular and context-free operands.

1. Introduction

The *binary word operations*, whose simplest examples are catenation and left/right quotient, have been extensively studied in the formal language theory. They are important for composition/decomposition of languages and their descriptions (grammars, automata). They are also of key importance for forming algebraic structures of formal languages, as the abstract families of languages (AFL) [19].

* Corresponding author. Institute of Computer Science, Silesian University, 74601 Opava, Czech Republic.
E-mail addresses: lila@csd.uwo.ca (L. Kari), sosik@csd.uwo.ca, petr.sosik@fpf.slu.cz (P. Sosík).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.09.038

Among the basic topics connected with them, besides their closure properties, we account *language* (*in*)*equations* involving these operations, see e.g. [20,14]. Various related problems have been studied during the last two decades, see e.g. [5–7]. The scope of the studied sequential insertion/deletion operations include insertion, shuffle, literal shuffle, deletion, bipolar deletion, scattered deletion, as well as their iterated and regulated versions. Recently, the applications of such language equations were shown in the coding theory for modelling noisy channels [8] or in the biocomputing research for characterization of sets of codewords for DNA computing [9,10].

Many of the mentioned insertion/deletion operations share the same principle, while they differ in positions where the letters of one argument are inserted/deleted into/from another one. Then one can characterize all these positions by a set of binary strings called the *trajectories*. Trajectories were introduced in [18] for a class of *insertion* operations, and their closure properties, algebraic properties and applications to concurrent processes modelling were studied. Further related problems were addressed, e.g. in [15–17]. The key concept of *shuffle on trajectories* involves many common word operations as its special cases, hence it allows to produce general results valid for the whole class of operations. Its importance and utility became more obvious in 2003 when the inverse operation, the *deletion on trajectories* was independently introduced in [1,13]. Several new results have been obtained since these two reports, some rather theoretical [2–4], some involving applications [9–11].

The paper is organized as follows. In Section 2 we fix some elementary notation of formal language theory. In Section 3 we recall definitions from [1,13,18], introducing the operations on trajectories. We give their characterization and some closure properties in Section 4. In Section 5 we study certain algebraic properties of deletion on trajectories, which differ substantially from the insertion case.

The key Section 6 deals with linear language equations of the form $L_1 \diamond L_2 = R$, $\diamond$ being an insertion or deletion operation. Particularly we focus on the problem whether $L_1 \diamond L_2 = R$ holds or not, given the languages L_1, L_2, R, R are regular. The problem can be easily shown to be decidable for all the studied operations when L_1, L_2 are regular. More interesting is the case when one of L_1, L_2 is context-free. We give an exact characterization of the class of sets of trajectories (and hence the class of binary word operations) for which this problem is decidable and for which it is not. This characterization generalizes several previous studies, see [5–7], and brings also some new results as its special cases.

2. Definitions and preliminaries

An *alphabet* is a finite and nonempty set of symbols. In the sequel we shall use a fixed alphabet Σ. Σ is assumed to be non-singleton, if not stated otherwise. The set of all words (over Σ) is denoted by Σ^*. This set includes the *empty word* λ. The length of a word w is denoted by $|w|$. $|w|_x$ denotes the number of non-overlapping occurrences of x within w, for $w \in \Sigma^*$, $x \in \Sigma^+$. For a nonnegative integer n and a word w, we use w^n to denote the n concatenated copies of w.

A mapping $\alpha : \Sigma^* \to 2^{\Sigma^*}$ is called a *finite substitution* of Σ^* if $\alpha(uv) = \alpha(u)\alpha(v)$ for all $u, v \in \Sigma^*$, and $\alpha(a)$ is finite for all $a \in \Sigma$. Moreover, if $|\alpha(a)| = 1$ for all $a \in \Sigma$, then α is called a *morphism* of Σ^*.

A language L is a set of words, or equivalently a subset of Σ^*. If n is a nonnegative integer, we write L^n for the language consisting of all words of the form $w_1 \cdots w_n$ such that each w_i is in L. We also write L^* for the language $L^0 \cup L^1 \cup L^2 \cup \cdots$ and L^+ for the language $L^* - \{\lambda\}$. The notation L^c represents the complement of the language L, that is, $L^c = \Sigma^* - L$. For the classes of regular, context-free, and context sensitive languages, we use the notations REG, CF and CS, respectively.

A *finite transducer* (in standard form) is a sextuple $T = (S, \Sigma, \Sigma', s_0, F, P)$ such that S is the finite and nonempty set of states, s_0 is the start state, F is the set of final states, and the set P consists of productions of the form $sx \to yt$ where s and t are states in S, $x \in \Sigma \cup \{\lambda\}$ and $y \in \Sigma' \cup \{\lambda\}$. If x is nonempty for every production then the transducer is called a generalized sequential machine (gsm). The *relation* realized by the transducer T is denoted by $R(T)$. We refer the reader to [19] for further details on automata and formal languages.

A *binary word operation* is a mapping $\diamond\colon \Sigma^* \times \Sigma^* \to 2^{\Sigma^*}$, where 2^{Σ^*} is the set of all subsets of Σ^*. For any languages X and Y over Σ, we define

$$X \diamond Y = \bigcup_{u \in X, v \in Y} u \diamond v. \tag{1}$$

The word operation $\diamond'$ defined by $u \diamond' v = v \diamond u$ is called *reversed* $\diamond$. For many examples of common binary word operations we refer the reader to [6,7,14,18], and also to the next section. Here we recall only two of them.

Shuffle (or scattered insertion): $u \sqcup\!\sqcup v = \{u_1 v_1 \cdots u_k v_k u_{k+1} \mid k \geq 1,$
$u = u_1 \cdots u_k u_{k+1}, v = v_1 \cdots v_k\}$.

Scattered deletion: $u \rightsquigarrow v = \{u_1 \cdots u_k u_{k+1} \mid k \geq 1, u = u_1 v_1 \cdots u_k v_k u_{k+1}, v = v_1 \cdots v_k\}$.

3. Shuffle and deletion on trajectories

The shuffle on trajectories was introduced in [18] as a concept generalizing the above insertion word operations. *Trajectory* is essentially a syntactical condition specifying how two words α and β are merged together into a resulting word. Formally, a trajectory is a string over the *trajectory alphabet* $V = \{0, 1\}$. The following definition and the example are due to [18]. Let Σ be an alphabet and let t be a trajectory, $t \in V^*$. Let α, β be two words over Σ.

Definition 3.1. The shuffle of α with β on the trajectory t, denoted by $\alpha \sqcup\!\sqcup_t \beta$, is defined as follows:

$$\alpha \sqcup\!\sqcup_t \beta = \{\alpha_1\beta_1 \ldots \alpha_k\beta_k \mid \alpha = \alpha_1 \ldots \alpha_k,\ \beta = \beta_1 \ldots \beta_k,\ t = 0^{i_1}1^{j_1} \ldots 0^{i_k}1^{j_k},\ \text{where}$$
$$|\alpha_m| = i_m \text{ and } |\beta_m| = j_m \text{ for all } m,\ 1 \leqslant m \leqslant k\}.$$

Observe that in $\alpha \sqcup\!\sqcup_t \beta$, the positions of 0's in t correspond to letters of α, while 1's correspond to letters of β.

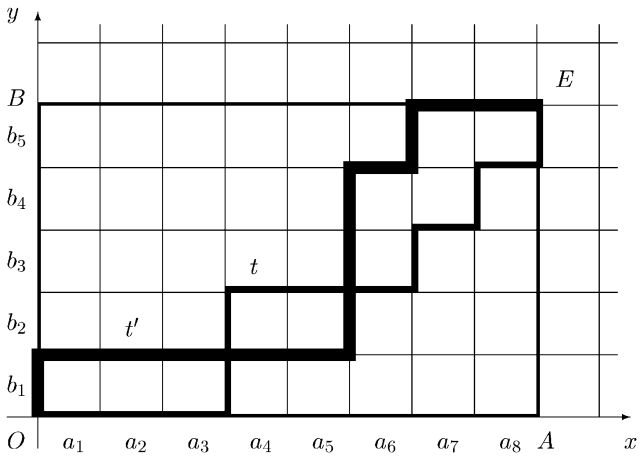

Fig. 1. Geometric representation of the shuffle on trajectories. The figure reprinted from [18] with permission of the authors.

Example 3.2. Let α and β be the words $\alpha = a_1a_2 \dots a_8$, $\beta = b_1b_2 \dots b_5$ and assume that $t = 0^3 1^2 0^3 10101$. The shuffle of α and β on the trajectory t is

$$\alpha \sqcup\!\sqcup_t \beta = \{a_1a_2a_3b_1b_2a_4a_5a_6b_3a_7b_4a_8b_5\}.$$

The shuffle operation has a natural two-dimensional geometric representation. The trajectory t defines a line starting in the origin and continuing one unit to the right or up, depending on the symbols in t. The symbol 0 stands for the *right* direction and 1 stands for the *up* direction, see Fig. 1. The thinner line corresponds to the above trajectory t, the bolder one to another trajectory $t' = 10^5 1^3 010^2$.

Trajectory conditions can be applied also in the *scattered deletion* case. The following definition has been introduced independently in the reports [1,13].

Definition 3.3. The deletion of β from α on trajectory t is the following binary word operation:

$$\alpha \rightsquigarrow_t \beta = \{\alpha_1 \dots \alpha_k \mid \alpha = \alpha_1\beta_1 \dots \alpha_k\beta_k,\ \beta = \beta_1 \dots \beta_k,\ t = 0^{i_1}1^{j_1} \dots 0^{i_k}1^{j_k},\ \text{where } |\alpha_m| = i_m \text{ and } |\beta_m| = j_m \text{ for all } m,\ 1 \leqslant m \leqslant k\}.$$

It follows from the above definition that if $|\alpha| \neq |t|$ or $|\beta| \neq |t|_1$, then $\alpha \rightsquigarrow_t \beta = \emptyset$. Observe that $\alpha \sqcup\!\sqcup_t \beta = \gamma$ iff $\gamma \rightsquigarrow_t \beta = \alpha$.

A *set of trajectories* is any set $T \subseteq V^*$. We extend the operations $\sqcup\!\sqcup_t$, $\rightsquigarrow_t$ to sets of trajectories as follows:

$$\alpha \sqcup\!\sqcup_T \beta = \bigcup_{t \in T} \alpha \sqcup\!\sqcup_t \beta, \quad \alpha \rightsquigarrow_T \beta = \bigcup_{t \in T} \alpha \rightsquigarrow_t \beta, \tag{2}$$

where $\alpha, \beta \in \Sigma^*$ and $T \subseteq V^*$. For extension to languages principle (1) holds.

Example 3.4. The following binary word insertion and deletion operations are special cases of shuffle and deletion on trajectories:

1. Let $T = 0^*1^*$, then $\sqcup\!\sqcup_T = \cdot$, the catenation operation, and $\rightsquigarrow_T = \longrightarrow_{\mathrm{rq}}$, the right quotient. For $T = 1^*0^*$, one gets $\rightsquigarrow_T = \longrightarrow_{\mathrm{lq}}$, the left quotient.
2. Let $T = 0^*1^*0^*$, then $\sqcup\!\sqcup_T = \longleftarrow$, the insertion, and $\rightsquigarrow_T = \longrightarrow$, the deletion. For $T = 1^*0^*1^*$, $\rightsquigarrow_T = \rightleftharpoons$, the dipolar deletion.
3. Let $T = \{0, 1\}^*$, then $\rightsquigarrow_T = \sqcup\!\sqcup$, the shuffle, and $\rightsquigarrow_T = \rightsquigarrow$, the scattered deletion.
4. Let $T = (01)^*(0^* \cup 1^*)$, then $\rightsquigarrow_T = \sqcup\!\sqcup_{\mathrm{l}}$, the literal shuffle, and $\rightsquigarrow_T = \rightsquigarrow_{\mathrm{l}}$, the literal deletion.
5. Let $T = (01)^*$, then $\rightsquigarrow_T = \sqcup\!\sqcup_{\mathrm{bl}}$, the balanced literal shuffle, and $\rightsquigarrow_T = \rightsquigarrow_{\mathrm{bl}}$, the balanced literal deletion.

4. Closure and characterization results

We recall first that the operations of shuffle and deletion on trajectories are mutual left inversions. See [1,13] for definitions of inversion operations and for further details.

Similarly as with shuffle on trajectories in [18], we can represent the deletion on trajectories via simpler language operations. The following two theorems give examples of such characterizations. Another characterization was given in [1].

Theorem 4.1. *For all languages L_1 and L_2, $L_1, L_2 \subseteq \Sigma^*$, and for all sets of trajectories $T \subseteq V^*$, there exists a gsm M and a letter-to-letter morphism h such that*

$$L_1 \rightsquigarrow_T L_2 = (M(L_1 \sqcup\!\sqcup T) \rightsquigarrow h(L_2)) \cap \Sigma^*. \tag{3}$$

Proof. Let $\Sigma_1 = \{a_1 \mid a \in \Sigma\}$ be a copy of Σ such that Σ, Σ_1 and V are pairwise disjoint alphabets. Let further $h : \Sigma \longrightarrow \Sigma_1$ be a morphism such that $h(a) = a_1$, $a \in \Sigma$. Consider finally the gsm $M = (Q, \Sigma', \Delta, q_0, \delta, F)$, where $Q = \{q_0, q_1, q_2\}$, $\Sigma' = \Sigma \cup \Sigma_1 \cup V$, $\Delta = \Sigma \cup \Sigma_1$, $F = \{q_0\}$ and

$$\delta(q_0, 1) = (q_1, \lambda), \quad \delta(q_0, 0) = (q_2, \lambda),$$
$$\delta(q_1, a) = (q_0, a_1), \quad \delta(q_2, a) = (q_0, a)$$

for all $a \in \Sigma$.

The gsm M accepts only the words from $T \sqcup\!\sqcup_{\mathrm{bl}} L_1$, all other words from $L_1 \sqcup\!\sqcup T$ are rejected. Let $t \sqcup\!\sqcup_{\mathrm{bl}} x$, $x \in L_1$, $t \in T$ be an accepted word, then M rewrites all its substrings $1a$ to a_1, and all its substrings $0a$ to a. Hence within the result $y = M(t \sqcup\!\sqcup_{\mathrm{bl}} x)$, the letters from Σ_1 are exactly those that should be deleted from x via $\rightsquigarrow_t$ (we call them "marked for deletion").

Now, words from $h(L_2)$ can be deleted from y via the operation $y \rightsquigarrow h(L_2)$. Due to the intersection with Σ, the results contributes to the right-hand side of (3) iff the deleted letters were all those marked for deletion, and hence the result is equal to $x \rightsquigarrow_t z$ for some $z \in L_2$. We can conclude that (3) holds. □

Theorem 4.2. *For all languages L_1 and L_2, $L_1, L_2 \subseteq \Sigma^*$, and for all sets T of trajectories, there exists a regular set R and morphisms h_1, h_2 and h_3 such that*

$$L_1 \rightsquigarrow_T L_2 = h_3(((L_1 \sqcup\!\sqcup h_1(T)) \cap R) \rightsquigarrow h_2(L_2)) \cap \Sigma^*. \tag{4}$$

Proof. Consider the following morphisms:

$h_1 : V \longrightarrow V^*$, where $h_1(0) = 0$, $h_1(1) = 11$,
$h_2 : \Sigma \longrightarrow (\Sigma \cup V)^*$, where $h_2(a) = 1a1$, $a \in \Sigma$,
$h_3 : (\Sigma \cup V) \longrightarrow (\Sigma \cup V)^*$, where $h_3(a) = a$, $a \in \Sigma$, $h_3(1) = 1$, $h_3(0) = \lambda$.

Let further $R = (0\Sigma \cup 1\Sigma 1)^*$. The principle of the construction is similar to that of Theorem 4.1 and is left to the reader. □

The following closure properties of shuffle on trajectories are known to hold [18].

Lemma 4.3. *Let T be a set of trajectories. The following assertions are equivalent*:
(i). *For all regular languages L_1, L_2, $L_1 \sqcup\!\sqcup_T L_2$ is a regular language.*
(ii). *T is a regular language.*

A similar result for the deletion case was independently shown in [1,13].

Lemma 4.4. *For all regular languages L_1, L_2, and a regular set of trajectories T, $L_1 \rightsquigarrow_T L_2$ is a regular language.*

Interestingly enough, the statement analogous to Lemma 4.3 does not hold for the deletion on trajectories, as there are non-regular sets of trajectories T such that $L_1 \rightsquigarrow_T L_2$ is regular for all regular languages L_1, L_2. Details of this problem are studied in [1]. We also refer to [1,13,18] for further closure properties of insertion and deletion on trajectories.

5. Algebraic properties

Various algebraic properties as completeness and determinism of sets of trajectories, commutativity, associativity and distributivity have been studied for shuffle of trajectories [18]. Here we give an analogous study for the case of deletion on trajectories.

Definition 5.1. A set of trajectories T is called *deterministic* iff $\mathrm{card}(\alpha \sqcup\!\sqcup_T \beta) \leqslant 1$, for all $\alpha, \beta \in \Sigma^*$.

Theorem 5.2. *If a set of trajectories T is deterministic, then* $\mathrm{card}(\alpha \rightsquigarrow_T \beta) \leqslant 1$, *for all $\alpha, \beta \in \Sigma^*$, but not conversely.*

Proof. (i) Assume that $\mathrm{card}(\alpha \rightsquigarrow_T \beta) \geqslant 2$ for some $\alpha, \beta \in \Sigma^*$. Then there are $t_1, t_2 \in T$ such that $t_1 \neq t_2$, $|t_1|_0 = |t_2|_0$ and $|t_1|_1 = |t_2|_1$. Consider $\Sigma = \{a, b\}$ and let $g : V^* \longrightarrow \Sigma^*$ be a morphism defined by $g(0) = a$ and $g(1) = b$. Denote $i = |t_1|_0$, $j = |t_1|_1$.

Then $\{g(t_1), g(t_2)\} \subseteq a^i \sqcup\!\sqcup_T b^j$. As g is a bijection, $g(t_1) \neq g(t_2)$ and hence T is not deterministic.

(ii) Let $T = \{01, 10\}$. On the one hand, T is not deterministic as $a \sqcup\!\sqcup b = \{ab, ba\}$. Suppose, on the other hand, that $\mathrm{card}(\alpha \rightsquigarrow_T \beta) \geqslant 2$ for some $\alpha, \beta \in \Sigma^*$. Then $\beta \in \Sigma$, hence $\alpha = \beta\beta$ and $\alpha \rightsquigarrow_T \beta = \{\beta\}$, a contradiction. □

If we consider *distributivity of* $\rightsquigarrow_T$ *over union*, then the reader can easily check that it holds true both on the left-hand and on the right-hand side.

The situation is different considering *commutativity* and *associativity*. Intuitively, for all but very special sets of trajectories T, the operation $\rightsquigarrow_T$ is neither commutative nor associative, due to asymmetrical role of its arguments. All the deletion operations mentioned in Sections 2 and 3 fall into this category.

Theorem 5.3. *For a set of trajectories T, the following two assertions are equivalent*:
(i) $T \subseteq \{1\}^*$.
(ii) $\rightsquigarrow_T$ *is a commutative operation, i.e.* $L_1 \rightsquigarrow_T L_2 = L_2 \rightsquigarrow_T L_1$ *for all* $L_1, L_2 \subseteq \Sigma^*$.

Proof. (i) Let $T \subseteq \{1\}^*$ be a set of trajectories, then

$$L_1 \rightsquigarrow_T L_2 = \begin{cases} \{\lambda\} & \text{if } 1^{|w|} \in T \text{ for some } w \in L_1 \cap L_2, \\ \emptyset & \text{otherwise,} \end{cases}$$

for any languages L_1, L_2, and hence $\rightsquigarrow_T$ is commutative.

(ii) Let there be $t \in T$ such that $|t|_0 > 0$. Then for the languages $L_1 = \{a^{|t|}\}$, $L_2 = \{a^{|t|_1}\}$ we have $L_1 \rightsquigarrow_T L_2 = \{a^{|t|_0}\}$, $L_2 \rightsquigarrow_T L_1 = \emptyset$ and $\rightsquigarrow_T$ is not commutative. □

Corollary 5.4. $\rightsquigarrow_T$ *is a commutative operation iff* $L_1 \rightsquigarrow_T L_2 \subseteq \{\lambda\}$ *for all languages* $L_1, L_2 \in \Sigma^*$.

Theorem 5.5. *For a set of trajectories T, the following two assertions are equivalent*:
(i) *For all* $t_1 \in (1^m \sqcup\!\sqcup 0^n)$, $t_2 \in (1^n \sqcup\!\sqcup 0^i)$, $t_3 \in (1^j \sqcup\!\sqcup 0^{m+n})$, $i, j, m, n \in \mathbb{N}$,

$$m > 0 \text{ and } t_1 \in T \text{ implies } t_2 \notin T, \quad \text{(a)}$$
$$m > 0 \text{ and } t_1 \in T \text{ implies } t_3 \notin T, \quad \text{(b)}$$
$$t_1 \in T \text{ and } 0^m \in T \text{ implies } 0^n \in T, \quad \text{(c)}$$
$$t_1 \in T \text{ and } 0^n \in T \text{ implies } 0^m \in T. \quad \text{(d)}$$

(ii) $\rightsquigarrow_T$ *is an associative operation, i.e.* $(L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3 = L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3)$ *for all* $L_1, L_2, L_3 \subseteq \Sigma^*$.

Proof. (i) Consider languages $L_1, L_2, L_3 \subseteq \Sigma^*$ and a set of trajectories T satisfying (i). Let $y \in ((L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3)$, then there are $t_1, t_2 \in T$, $x_1 \in L_1$, $x_2 \in L_2$ and $x_3 \in L_3$ such that $(x_1 \rightsquigarrow_{t_1} x_2) \rightsquigarrow_{t_2} x_3 = \{y\}$.

Let $t_2 \in (1^m \sqcup\!\sqcup 0^n)$ for some $m, n \geqslant 0$, then $t_1 \in (1^k \sqcup\!\sqcup 0^{m+n})$ for some $k \geqslant 0$. We can deduce that $m = 0$ and hence also $x_3 = \lambda$, because for $m > 0$, $t_2 \in T$ would imply $t_1 \notin T$ due to (b), a contradiction. As $t_1 \in (1^k \sqcup\!\sqcup 0^n)$ and $t_2 = 0^n$, due to (d) also $0^k \in T$ and it follows that $\{y\} = (x_1 \rightsquigarrow_{t_1} x_2) \rightsquigarrow_{t_2} x_3 = x_1 \rightsquigarrow_{t_1} x_2 = x_1 \rightsquigarrow_{t_1} (x_2 \rightsquigarrow_{0^k} x_3) \subseteq L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3)$.

We have shown that $(L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3 \subseteq L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3)$. Using analogous arguments for (a) and (c) we can show also that $L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3) \subseteq (L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3)$ which together assures the associativity of $\rightsquigarrow_T$.

(ii) We show that violation of any of the conditions (a)–(d) would make T non-associative.

(a) Let there be $t_1, t_2 \in T$, $t_1 \in (1^m \sqcup\!\sqcup 0^n)$, $t_2 \in (1^n \sqcup\!\sqcup 0^i)$, for some $m > 0$, $n, i \geqslant 0$. Consider the word $w \in \{a, b\}^*$, such that $y = \phi(t_1)$, where $\phi(0) = a$, $\phi(1) = b$. Then

$$a^{n+i} \rightsquigarrow_T (y \rightsquigarrow_T b^m) = a^{n+i} \rightsquigarrow_{t_2} (y \rightsquigarrow_{t_1} b^m) = \{a^i\}$$

and

$$(a^{n+i} \rightsquigarrow_T y) \rightsquigarrow_T b^m = \emptyset \rightsquigarrow_T b^m = \emptyset,$$

as y contains at least one b. Hence $\rightsquigarrow_T$ is not associative for the languages $\{a^{n+i}\}$, $\{y\}$ and $\{b^m\}$.

(b) The proof is analogous to (a) and is left to the reader.

(c) Let there be $t_1 \in T$, $t_1 \in (1^m \sqcup\!\sqcup 0^n)$, and let further $0^m \in T$ and $0^n \notin T$. Then

$$a^{n+m}(\rightsquigarrow_T a^m \rightsquigarrow_T \lambda) = a^{n+m} \rightsquigarrow_{t_1} (a^m \rightsquigarrow_{0^m} \lambda) = \{a^n\},$$
$$(a^{n+m} \rightsquigarrow_T a^m) \rightsquigarrow_T \lambda = a^n \rightsquigarrow_T \lambda = \emptyset,$$

and hence $\rightsquigarrow_T$ is non-associative.

(d) Analogous to (c).

□

Corollary 5.6. *If $\rightsquigarrow_T$ is associative, then $(L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3 = \emptyset = L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3)$ for all $L_1, L_2 \subseteq \Sigma^*$, $L_3 \subseteq \Sigma^+$.*

Proof. As it is shown in the proof of Theorem 5.5, part (i), if $\rightsquigarrow_T$ is associative and $(x_1 \rightsquigarrow_T x_2) \rightsquigarrow_T x_3 \neq \emptyset$ for some $x_1, x_2, x_3 \in \Sigma^*$, then $x_3 = \lambda$. Hence $(x_1 \rightsquigarrow_T x_2) \rightsquigarrow_T x_3 = \emptyset$ for all $x_1, x_2 \in \Sigma^*$, $x_3 \in \Sigma^+$, and thus also $(L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3 = \emptyset$ for all $L_1, L_2 \subseteq \Sigma^*$, $L_3 \subseteq \Sigma^+$.

Furthermore, if $\rightsquigarrow_T$ is associative and $L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3) \neq \emptyset$ for some $L_1, L_2 \subseteq \Sigma^*$, $L_3 \subseteq \Sigma^+$, then $(L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3 \neq L_1 \rightsquigarrow_T (L_2 \rightsquigarrow_T L_3)$, a contradiction, since $(L_1 \rightsquigarrow_T L_2) \rightsquigarrow_T L_3 = \emptyset$ as we have just shown. □

6. Decision problems

In this section we study three elementary types of decision problems for language equations involving the shuffle and deletion on trajectories. They are formulated generally for an arbitrary binary language operation $\diamond$ in accordance with [6].

Q_0 : For given languages L_1, L_2, R, R regular, is $L_1 \diamond L_2 = R$?

Q_1 : For given languages L_2, R, R regular, does there exists $X \subseteq \Sigma^*$ such that $X \diamond L_2 = R$?

Q_2 : For given languages L_1, R, R regular, does there exists $X \subseteq \Sigma^*$ such that $L_1 \diamond X = R$?

The variant of problem Q_0 for a singleton language $L_2 = \{w\}$ is denoted by Q_0^w. Similarly, the variants of the problems Q_1 and Q_2 for a singleton language $X = \{w\}$ are denoted by Q_1^w and Q_2^w, respectively.

6.1. The regular case

Let us focus on the case when L_1, L_2 and T are all regular first. Then $L_1 \sqcup\!\sqcup_T L_2$, $L_1 \rightsquigarrow_T L_2$ are also regular languages by Lemmata 4.3 and 4.4. Hence the problems Q_0 and Q_0^w are decidable. The following theorem addressing the problems Q_1, Q_1^w and Q_2 show that in this case these are also decidable. The results were independently proven in [1,13].

Theorem 6.1. *Let L_1, L_2, R be regular languages and T a regular set of trajectories. The following problems are decidable*:

(i) *"Does there exist a solution X to the equation $L_1 \sqcup\!\sqcup_T X = R$ ($L_1 \rightsquigarrow_T X = R$)?"* (*problem Q_2*);

(ii) *"Does there exist a solution X to the equation $X \sqcup\!\sqcup_T L_2 = R$ ($X \rightsquigarrow_T L_2 = R$)?"* (*problem Q_1*);

(iii) *"Does there exist a word w such that $w \sqcup\!\sqcup_T L_2 = R$ ($w \rightsquigarrow_T L_2 = R$)?"* (*problem Q_1^w*).

As a consequence, the problems Q_1, Q_1^w and Q_2 are decidable for all the binary word operations mentioned in Sections 2 and 3, as they are special cases of $\sqcup\!\sqcup_T$ or $\rightsquigarrow_T$. For a majority of them, however, this was already known [5–7].

The problem Q_2^w is decidable for the operation $\sqcup\!\sqcup_T$ due to the fact that $x \sqcup\!\sqcup_T y = y \sqcup\!\sqcup_{\phi(T)} x$, where ϕ is a coding such that $\phi(0) = 1$ and $\phi(1) = 0$. Hence it can be easily reduced to the problem Q_1^w. In case of the operation $\rightsquigarrow_T$, however, such a reduction is not possible and the decision status of the problem Q_2^w remains open.

6.2. The context-free case

Now we address the problems Q_0, Q_1, Q_2 in the case when at least one of the involved languages is context-free (and the remaining ones are regular).

Generally, for context-free languages L_1, L_2, a regular language R and a regular set of trajectories T, the problems Q_1, Q_1^w, Q_2 and Q_2^w are all undecidable. We refer to [6,12] where the undecidability of the mentioned problems is shown for the operations under consideration, namely $\cdot$, $\leftarrow$, $\sqcup\!\sqcup$, $\rightarrow$, $\rightarrow_{\mathrm{lq}}$, $\rightarrow_{\mathrm{rq}}$, $\rightsquigarrow$. The only exception is the case of $\rightsquigarrow$ for which the problem Q_1^w is rather surprisingly decidable [12]. Further details can also be found in [3].

Consider now the problem Q_0 and assume just one of L_1 or L_2 to be context free and the other one regular. The former case (L_1 context free) has been studied in [6,7], where its undecidability is shown for the six above-mentioned operations (cf. Corollary 6.9). The latter case seems to be rather unexploited yet. In the rest of this section, we study both these subcases of the problem Q_0. For both of them we characterize precisely the class of the trajectory sets (and hence the class of binary operations) for which these problems are (un)decidable. In the following lemma, we call an integer set $D \subseteq \mathbb{N}$ *regular* iff it is a length set of a regular language.

Lemma 6.2. *Consider an arbitrary but fixed infinite regular integer set D. Denote $R_D = \bigcup_{d\in D}\{0,1\}^d$, a regular language over the alphabet $\{0,1\}$. Then the problem "Does $R_D \subseteq L$ hold true" is undecidable for a context-free language L.*

Proof. Denote by $(\mathcal{U};\mathcal{V}) = (u_1,\ldots,u_n; v_1,\ldots,v_n)$, $n \geqslant 1$, an instance of post correspondence problem (PCP) over $\{0,1\}$, and let $k = \lceil \log_2 n \rceil + 2$. Consider the languages

$$L_{\mathcal{U}} = 1^*\{(i_m)_2 \ldots (i_1)_2 0^k u_{i_1} \ldots u_{i_m} \mid m > 0,\ 1 \leqslant i_j \leqslant n \text{ for all } i_j,\ 1 \leqslant j \leqslant m\},$$
$$L_{\mathcal{V}} = 1^*\{(i_m)_2 \ldots (i_1)_2 0^k v_{i_1} \ldots v_{i_m} \mid m > 0,\ 1 \leqslant i_j \leqslant n \text{ for all } i_j,\ 1 \leqslant j \leqslant m\},$$

where $(i_j)_2$ denotes the k-digits binary representation of i_j. Observe that each string $(i_j)_2$ starts with 0 and contains at least one 1, so that none of them equals to 0^k. Let $L = L^c_{\mathcal{U}} \cup L^c_{\mathcal{V}}$. One can easily check that L is a context-free language and that $L^c = L_{\mathcal{U}} \cap L_{\mathcal{V}}$ is empty iff there is no solution $(i_1,\ldots,i_m)$ of $(\mathcal{U};\mathcal{V})$. We show that $R_D \subseteq L$ holds true iff the instance $(\mathcal{U};\mathcal{V})$ of PCP has no solution.

- If $(\mathcal{U};\mathcal{V})$ has no solution, then $L = \Sigma^*$ and $R_D \subseteq L$ holds true.
- Assume that $(\mathcal{U};\mathcal{V})$ has a solution $(i_1,\ldots,i_m)$ for some $m \geqslant 1$, and denoted by $\ell = |(i_m)_2 \ldots (i_1)_2 0^k u_{i_1} \ldots u_{i_m}|$ the length of the shortest word in L^c corresponding to that solution. Then for each $d \geqslant \ell$ there exists a word $x \in L^c$ of the length d thanks to the prefix 1^* of the language L^c. Choose a number $d \in D$ such that $d \geqslant \ell$, then there is a word $x \in R_D$, $|x| = d$, such that $x \in L^c$, hence $x \notin L$ and $R_D \subseteq L$ does not hold. □

We address the variant of the problem Q_0 when the left operand is context-free first. For a set of trajectories $T \subseteq V^*$ and $a \in V$, denoted by $\Psi_a(T) \subseteq \mathbb{N}$ the Parikh image of T restricted to a,

$$\Psi_a(T) = \{|t|_a \mid t \in T\}.$$

Considering an alphabet Σ, denote further $R_0(T) = \bigcup_{d\in\Psi_0(T)} \Sigma^d$. Obviously, if T is regular, then also $R_0(T)$ is a regular language which can be obtained from T by a finite substitution $\phi(0) = \Sigma$, $\phi(1) = \lambda$.

Theorem 6.3. *Let T be an arbitrary but fixed regular set of trajectories. The problem "Is $L_1 \sqcup\!\sqcup_T L_2 = R$" is decidable for a context-free language L_1 and regular languages L_2, R if and only if $\Psi_0(T)$ is finite.*

Proof. (i) Consider a regular set of trajectories T such that $\Psi_0(T)$ is finite. Observe that $L_1 \sqcup\!\sqcup_T L_2 = (L_1 \cap R_0(T)) \sqcup\!\sqcup_T L_2$. As $R_0(T)$ is an effectively constructible finite language, the same holds for $L_1 \cap R_0(T)$. Hence the language $L_1 \sqcup\!\sqcup_T L_2$ is regular and it is decidable whether or not $L_1 \sqcup\!\sqcup_T L_2 = R$ holds for a regular language R.

(ii) Assume that $\Psi_0(T)$ is infinite. Consider $\Sigma = \{0,1\}$ and let $L \subseteq \Sigma^*$ be an arbitrary context-free language. Consider further the regular language

$$R = \{0,1\}^* \sqcup\!\sqcup_T c^* = R_0(T) \sqcup\!\sqcup_T c^*$$

over the alphabet $\{0,1,c\}$. We show that

$$L \sqcup\!\sqcup_T c^* = R \text{ iff } R_0(T) \subseteq L. \tag{5}$$

(a) If $R_0(T) \subseteq L$, then

$$L \sqcup\!\sqcup_T c^* = (L \cap R_0(T)) \sqcup\!\sqcup_T c^* = R_0(T) \sqcup\!\sqcup_T c^* = R.$$

(b) If $R_0(T) \not\subseteq L$, then there is a $w \in R_0(T)$ such that $w \notin L$. Then the set $w \sqcup\!\sqcup_T c^*$ is nonempty and for all $w' \in w \sqcup\!\sqcup_T c^*$,

$$w' \in R_0(T) \sqcup\!\sqcup_T c^* \text{ and } w' \notin L \sqcup\!\sqcup_T c^*,$$

hence $w' \in R - (L \sqcup\!\sqcup_T c^*)$ and $L \sqcup\!\sqcup_T c^* \neq R$.

Now, if we could decide whether $L \sqcup\!\sqcup_T c^* = R$ holds or not, then we could due to (5) also decide whether $R_0(T) \subseteq L$ holds, which contradicts Lemma 6.2. □

Theorem 6.4. *Let T be an arbitrary but fixed regular set of trajectories. The problem "Is $L_1 \rightsquigarrow_T L_2 = R$" is decidable for a context-free language L_1 and regular languages L_2, R if and only if the set $\Psi_0(T)$ is finite.*

Proof. (i) For a context-free language L_1 and a regular language L_2, the language $L_1 \rightsquigarrow_T L_2$ is context-free due to the closure of the class *CF* under deletion on trajectories with regular languages, see [1,13]. Moreover, we can write

$$L_1 \rightsquigarrow_T L_2 = (L_1 \rightsquigarrow_T L_2) \cap R_0(T).$$

Assuming that $\Psi_0(T)$ is finite, the language $L_1 \rightsquigarrow_T L_2$ is also finite and we can effectively enumerate all its words. Hence the problem "Is $L_1 \rightsquigarrow_T L_2 = R$" is decidable.

(ii) Consider a regular set of trajectories T such that $\Psi_0(T)$ is infinite. Consider further $\Sigma = \{0, 1\}$ and let $L \subseteq \Sigma^*$ be an arbitrary context-free language. Let

$$L_1 = L \sqcup\!\sqcup_T c^*,$$
$$R = (\{0, 1\}^* \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^*$$

be languages over the alphabet $\{0, 1, c\}$. Observe that the definition of $\sqcup\!\sqcup_T$ implies that both

$$L_1 \subseteq R_0(T) \sqcup\!\sqcup_T c^*, \tag{6}$$

and

$$\{0, 1\}^* \sqcup\!\sqcup_T c^* = R_0(T) \sqcup\!\sqcup_T c^* \tag{7}$$

hold. We show that

$$L_1 \rightsquigarrow_T c^* = R \text{ iff } R_0(T) \subseteq L. \tag{8}$$

(a) If $R_0(T) \subseteq L$, then due to (6) and (7),

$$\begin{aligned} L_1 \rightsquigarrow_T c^* &= (L \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^* = ((L \cap R_0(T)) \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^* \\ &= (R_0(T) \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^* = (\{0, 1\}^* \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^* = R. \end{aligned}$$

(b) If $R_0(T) \not\subseteq L$, then there is a $w \in R_0(T)$ such that $w \notin L$. Then the set $W = (w \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^*$ is nonempty and for all $w' \in W$,

$$w' \in (R_0(T) \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^* \text{ and } w' \notin (L \sqcup\!\sqcup_T c^*) \rightsquigarrow_T c^*,$$

which can be due to (7) rewritten as

$$w' \in R - (L_1 \rightsquigarrow_T c^*),$$

and hence $L_1 \rightsquigarrow_T c^* \neq R$.

Hence, as in the previous proof, decidability of "$L_1 \rightsquigarrow_T c^* = R$" would imply decidability of "$R_0(T) \subseteq L$" which contradicts Lemma 6.2. □

Now we address the case when the right operand is context-free. We obtain the analogous result as in Theorem 6.3 for the case of $\sqcup\!\sqcup_T$ operation.

Theorem 6.5. *Let T be an arbitrary but fixed regular set of trajectories. The problem "Is $L_1 \sqcup\!\sqcup_T L_2 = R$" is decidable for a context-free language L_2 and regular languages L_1, R if and only if $\Psi_1(T) = \{|t|_1 \mid t \in T\}$ is finite.*

Proof. Observe that $\sqcup\!\sqcup_T = \sqcup\!\sqcup'_{\phi(T)}$ and $\Psi_1(T) = \Psi_0(\phi(T))$, where $\phi(0) = 1$ and $\phi(1) = 0$. Hence the statement follows by Theorem 6.3. □

For the operation $\rightsquigarrow_T$, however, the situation is different and needs a special attention due to the nonsymmetry of this operation. We recall one more notation of theory of formal languages. A language $T \subseteq V^*$ is *letter-bounded* if there is an $n \geqslant 0$ and $t_1, \ldots, t_n \in V$ such that $T \subseteq t_1^* t_2^* \ldots t_n^*$.

Theorem 6.6. *Let T be a regular set of trajectories which is not letter bounded. Then the problem "Is $L_1 \rightsquigarrow_T L_2 = R$" is undecidable for a context-free language L_2 and regular languages L_1, R.*

Proof. Observe first that if T is not letter bounded, then the integer set $\{|t|_{01} \mid t \in T\}$, characterizing the numbers of occurrences of the string 01 in T, is infinite. Denote

$$R_{01}(T) = \bigcup_{t \in T} \{0, 1\}^{|t|_{01}}.$$

The reader can easily check that $R_{01}(T)$ is a regular language. Let L be an arbitrary context-free language over an alphabet $\{0, 1\}$. Denote

$$L_1 = M(T \cdot \{c\}),$$
$$L_2 = (L \cap R_{01}(T)) \sqcup\!\sqcup c^*,$$
$$R = L_1 \rightsquigarrow_T (R_{01}(T) \sqcup\!\sqcup c^*),$$

where $M = (Q, V \cup c, \{0, 1, c\}, q_0, \delta, F)$ is a nondeterministic gsm, $Q = F = \{q_0, q_1\}$, and

$$\begin{aligned} &\delta(q_0, 0) = (q_1, \{\lambda\}), \quad \delta(q_1, 0) = (q_1, \{c\}),\\ &\delta(q_0, 1) = (q_0, \{c\}), \quad \delta(q_1, 1) = (q_0, \{a0, b1\}),\\ &\delta(q_0, c) = (q_0, \{c\}), \quad \delta(q_1, c) = (q_1, \{cc\}). \end{aligned}$$

One can observe that the language L_1 is produced from T replacing all the substrings 01 by either $a0$ or $b1$, and all the remaining symbols by c in each $t \in T$. We show that

$$L_1 \rightsquigarrow_T L_2 = R \quad \text{iff} \quad R_{01}(T) \subseteq L. \tag{9}$$

(i) If $R_{01}(T) \subseteq L$, then we have

$$L_1 \rightsquigarrow_T L_2 = L_1 \rightsquigarrow_T ((L \cap R_{01}(T)) \sqcup c^*) = L_1 \rightsquigarrow_T (R_{01}(T) \sqcup c^*) = R.$$

(ii) If $R_{01}(T) \not\subseteq L$, then there is a $w \in R_{01}(T)$ such that $w \notin L$. Observe that then there is a word $w' \in w \sqcup c^*$ such that after deleting w' via $\rightsquigarrow_T$ from L_1, the only symbols remaining in the result are a's, b's and c's, and moreover a's and b's are in the same order as 0's and 1's in w.

In other words, $w'' \in L_1 \rightsquigarrow_T w'$ holds for a word $w'' \in \phi(w) \sqcup c$, where $\phi : \{0, 1\}^* \longrightarrow \{a, b\}^*$ is a coding $\phi(0) = a$, $\phi(1) = b$. However, as $w \notin L$, there is no $x \in L_2$ such that $w'' \in L_1 \rightsquigarrow_T x$ would hold. Hence, we can write that

$$w'' \in L_1 \rightsquigarrow_T (R_{01}(T) \sqcup_T c^*) \quad \text{and} \quad w'' \notin L_1 \rightsquigarrow_T L_2,$$

which can be rewritten as

$$w'' \in R - (L_1 \rightsquigarrow_T L_2),$$

and hence $L_1 \rightsquigarrow_T L_2 \neq R$.

Now, if we could decide whether $L_1 \rightsquigarrow_T L_2 = R$ holds or not, then we could due to (9) also decide whether $R_{01}(T) \subseteq L$ holds, which contradicts Lemma 6.2. □

For a complementary statement, we benefit from the results given in [3].

Theorem 6.7. *Let T be a letter-bounded regular set of trajectories. Let further L be a context-free language and R a regular one. Then the language $R \rightsquigarrow_T L$ is regular and effectively constructible.*

Proof. By Corollary 1 in [3], if T is regular and letter bounded, then for an arbitrary L the language $R \rightsquigarrow_T L$ is regular. Moreover, there is a finite number of effectively constructible regular languages $R_1, \ldots, R_n$ such that $R \rightsquigarrow_T L = R_i$ for some i, $1 \leqslant i \leqslant n$.

Supposing that L is context-free, one can construct a context-free grammar G generating $R \rightsquigarrow_T L$, see Theorem 3.3 and Corollary 3.5 in [1] and their proofs. The algorithm computing

$R \rightsquigarrow_T L$ can now be outlined as follows.
1. Construct the set of languages $\mathcal{R} = \{R_i \mid L(G) \subseteq R_i,\ 1 \leqslant i \leqslant n\}$.
2. Choose an $\overline{R} \in \mathcal{R}$ such that $\overline{R} \subseteq R_i$ for all $R_i \in \mathcal{R}$. Such an $\overline{R}$ must always exist as $L(G) = R_i$ for some i, $1 \leqslant i \leqslant n$.
3. Output $\overline{R}$.

□

Combining the above two results, we can give the following answer to the open instances of the problem Q_0 for the operation $\rightsquigarrow_T$ with the right argument being context-free:

Theorem 6.8. *Let T be an arbitrary but fixed regular set of trajectories. The problem "Is $L_1 \rightsquigarrow_T L_2 = R$" is decidable for a context-free language L_2 and regular languages L_1, R if and only if T is letter-bounded.*

All the consequences of theorems in this section are summarized into a single table in the following corollary. Previously unpublished results are typed in boldface.

Corollary 6.9. *The following table holds true for the decision problem Q_0 : "Is $L_1 \diamond L_2 = R$?". The symbol D stands for a decidable, U for an undecidable problem*:

L_1	L_2	Operation $\diamond$											
		$\cdot$	$\longleftarrow$	$\sqcup\!\sqcup$	$\sqcup\!\sqcup_{\mathrm{l}}$	$\sqcup\!\sqcup_{\mathrm{bl}}$	$\xrightarrow[\mathrm{lq}]{}$	$\xrightarrow[\mathrm{rq}]{}$	$\longrightarrow$	$\rightleftharpoons$	$\rightsquigarrow$	$\rightsquigarrow_{\mathrm{l}}$	$\rightsquigarrow_{\mathrm{bl}}$
REG	*REG*	D	D	D	**D**	**D**	D	D	D	**D**	D	**D**	**D**
CF	*REG*	U	U	U	**U**	**U**	U	U	U	**U**	U	**U**	**U**
REG	*CF*	U	U	U	**U**	**U**	D	D	**D**	**D**	**U**	**U**	**U**

Acknowledgements

This research was supported by the Canada Research Chair Grant and NSERC Discovery Grant to L.K., and also by the Grant Agency of Czech Republic, Grant No. 201/02/P079 to P.S.

References

[1] M. Domaratzki, Deletion along trajectories, Tech. Report 2003-464, School of Computing, Queen's University, 2003; Theoret. Comput. Sci. 320 (2004) 293–313.
[2] M. Domaratzki, Splicing on routes versus shuffle and deletion along trajectories, Tech. Report 2003-471, School of Computing, Queen's University, 2003.
[3] M. Domaratzki, K. Salomaa, Decidability of trajectory-based equations, in: J. Fiala, V. Koubek, J. Kratochvíl (Eds.), Mathematical Foundations of Computer Science 2004, Lecture Notes in Computer Science, Vol. 3153, Springer, Berlin, 2004, pp. 723–734.
[4] M. Domaratzki, A. Mateescu, K. Salomaa, S. Yu, Deletion on trajectories and commutative closure, in: T. Harju, J. Karhumaki (Eds.), WORDS'03: Fourth Internat. Conf. on Combinatorics on Words, TUCS General Publication No. 27, August 2003, pp. 309–319.
[5] M. Ito, L. Kari, G. Thierrin, Shuffle and scattered deletion closure of languages, Theoret. Comput. Sci. 245 (2000) 115–133.

[6] L. Kari, On insertion and deletion in formal languages, Ph.D. Thesis, University of Turku, Finland, 1991.
[7] L. Kari, On language equations with invertible operations, Theoret. Comput. Sci. 132 (1994) 129–150.
[8] L. Kari, S. Konstantinidis, Language equations, maximality and error detection, J. Comput. System Sci., to appear.
[9] L. Kari, S. Konstantinidis, P. Sosík, Bond-free languages: formalizations, maximality and construction methods, in: C. Feretti, G. Mauri, C. Zandron (Eds.), DNA 10, Tenth Internat. Meeting on DNA Computing, Milano, University of Milano–Bicocca, 2004, pp. 16–25.
[10] L. Kari, S. Konstantinidis, P. Sosík, Preventing undesirable bonds between DNA codewords, in: C. Feretti, G. Mauri, C. Zandron (Eds.), DNA 10, Tenth Internat. Meeting on DNA Computing, Milano, University of Milano–Bicocca, 2004, pp. 375–384.
[11] L. Kari, S. Konstantinidis, P. Sosík, Substitutions on trajectories, in: J. Karhumäki, H. Maurer, G. Păun, G. Rozenberg (Eds.), Theory is Forever. Essays Dedicated to A. Salomaa on the Occasion of His 70th Birthday, Lecture Notes in Computer Science, Vol. 3113, Springer, Berlin, 2004, pp. 145–158.
[12] L. Kari, P. Sosík, On language equations with deletion, Bull. EATCS 83 (2004) 173–180.
[13] L. Kari, P. Sosík, Language deletion on trajectories, Dept. of Computer Science Tech. Report No. 606, University of Western Ontario, London, 2003.
[14] E.L. Leiss, Language Equations, Springer, New York, 1999.
[15] C. Martin-Víde, A. Mateescu, G. Rozenberg, A. Salomaa, Contexts on trajectories, TUCS Tech. Report No. 214, Turku Centre for Computer Science, 1998.
[16] A. Mateescu, A. Salomaa, Nondeterministic trajectories, Formal and natural computing: essays dedicated to Grzegorz Rozenberg, Lecture Notes in Computer Science, Vol. 2300, 2002, pp. 96–106.
[17] A. Mateescu, K. Salomaa, S. Yu, On fairness of manydimensional trajectories, J. Automata Languages and Combin. 5 (2000) 145–157.
[18] A. Mateescu, G. Rozenberg, A. Salomaa, Shuffle on trajectories: syntactic constraints, TUCS Tech. Report No. 41, Turku Centre for Computer Science, 1996; Theoret. Comput. Sci. 197 (1998) 1–56.
[19] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer, Berlin, 1997.
[20] A. Salomaa, Theory of Automata, Pergamon Press, Oxford, 1969.