Deletion along Trajectories
Technical Report 2003-464

Michael Domaratzki*
School of Computing, Queen's University,
Kingston, ON K7L 3N6.
email: domaratz@cs.queensu.ca

Abstract

We describe a new way to model deletions on formal languages, called *deletion along trajectories*. We examine its closure properties, and show that it serves as an inverse to shuffle on trajectories, recently introduced by Mateescu *et al.* This leads to results on the decidability of equations of the form $L \sqcup_T X = R$, where L, R are regular languages and X is unknown.

1 Introduction

Shuffle on trajectories, defined by Mateescu *et al.* [16] unifies operations which insert symbols of one word into another (see Section 2 for definitions). Among those operations in the literature generalized by shuffle on trajectory are concatenation, reverse and bi-concatenation, arbitrary, literal and perfect shuffles, and many others. This formalism has proven to be very powerful, and much work has recently been done on shuffle along trajectories (see, e.g., [4, 18, 19]).

Concurrent to this research, Kari and others [10, 11] have done research into the inverses of insertion-and shuffle-like operations, which have yielded decidability results for equations such as $XL = R$ where L, R are regular languages and X is unknown. The inverses of insertion- and shuffle-like operations are deletion-based operations such as *deletion*, *quotient*, *scattered deletion* and *bi-polar deletion* [10].

In this paper, we introduce the notion of *deletion along trajectories*, which is the equivalent of shuffle along trajectories for deletion-based operations. We show how it unifies operations such as deletion, quotient, scattered deletion and others. We also show how each shuffle operation based on a set of trajectories T has an inverse operation (both right and left inverse, see Section 5), defined by a deletion along a renaming of T. This yields the result that it is decidable whether equations of the form $L \sqcup_T X = R$ for regular languages L, T and R has a solution X.

*Research supported in part by an NSERC PGS-B graduate scholarship.

2 Definitions

For additional background in formal languages and automata theory, please see Yu [25]. Let Σ be a finite set of symbols, called *letters*. Then Σ^* is the set of all finite sequences of letters from Σ, which are called *words*. The empty word ϵ is the empty sequence of letters. The length of a word $w = w_1 w_2 \cdots w_n \in \Sigma^*$, where $w_i \in \Sigma$ is n, and is denoted $|w|$. Note that ϵ is the unique word of length 0.

A language L is any subset of Σ^*. By $\overline{L}$, we mean $\Sigma^* - L$, the complement of L.

A deterministic finite automaton (DFA) is a five-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to \Sigma$ is a transition function, $q_0 \in Q$ is the distinguished start state, and $F \subseteq Q$ are the final states. We extend δ to $Q \times \Sigma^*$ in the usual way. A word $w \in \Sigma^*$ is accepted by M if $\delta(q_0, w) \in F$. The language recognized by M, denoted $L(M)$ is the set of all strings recognized by M. A language is called regular if it is accepted by some DFA.

A nondeterministic finite automaton (NFA) is a five-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q, Σ, q_0 and F are in the deterministic case, while $\delta : Q \times (\Sigma \cup \epsilon) \to 2^Q$ is the nondeterministic transition function. Again, δ is extended to $Q \times \Sigma^*$ in the natural way. A word w is accepted by M if $\delta(q_0, w) \cap F \neq \emptyset$. It is known that the language accepted by an NFA is regular.

Given alphabets Σ, Δ, a morphism is a function $h : \Sigma^* \to \Delta^*$ satisfying $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$.

Shuffle on trajectories is defined by first defining the shuffle of two strings x and y over an alphabet Σ on a trajectory t, which is simply a string in $\{0, 1\}^*$.

If $x = ax'$ and $y = by'$ (with $a, b \in \Sigma$) then if $t = et'$, we have that

$$x \sqcup\!\sqcup_{et'} y = \begin{cases} a(x' \sqcup\!\sqcup_{t'} by') & \text{if } e = 0; \\ b(ax' \sqcup\!\sqcup_{t'} y') & \text{if } e = 1. \end{cases}$$

If $x = ax'$ ($a \in \Sigma$) and $y = \epsilon$, then

$$x \sqcup\!\sqcup_{et'} \epsilon = \begin{cases} a(x' \sqcup\!\sqcup_{t'} \epsilon) & \text{if } e = 0; \\ \emptyset & \text{otherwise.} \end{cases}$$

If $x = \epsilon$ and $y = by'$ ($b \in \Sigma$), then

$$\epsilon \sqcup\!\sqcup_{et'} y = \begin{cases} b(\epsilon \sqcup\!\sqcup_{t'} y') & \text{if } e = 1; \\ \emptyset & \text{otherwise.} \end{cases}$$

Finally, if $x = y = \epsilon$, then

$$\epsilon \sqcup\!\sqcup_{et'} \epsilon = \begin{cases} \epsilon & \text{if } et' = \epsilon \\ \emptyset & \text{otherwise.} \end{cases}$$

We extend shuffle on trajectories to sets $T \subseteq \{0, 1\}^*$ of trajectories as follows:

$$x \sqcup\!\sqcup_T y = \bigcup_{t \in T} x \sqcup\!\sqcup_t y.$$

Further, for $L_1, L_2 \subseteq \Sigma^*$, we define

$$L_1 \sqcup\!\sqcup_T L_2 = \bigcup_{x \in L_1} \bigcup_{y \in L_2} x \sqcup\!\sqcup_T y.$$

We now give our main definition, which models deletion operations with the addition of a set of trajectories. Let $x, y \in \Sigma^*$ be strings with $x = ax'$, $y = by'$ ($a, b \in \Sigma$). Let t be a string over $\{i, d\}$ such that $t = et'$ with $e \in \{i, d\}$. Then we define $x \rightsquigarrow_t y$ as follows:

$$x \rightsquigarrow_t y = \begin{cases} a(x' \rightsquigarrow_{t'} by') & \text{if } e = i \\ x' \rightsquigarrow_{t'} y' & \text{if } e = d \text{ and } a = b \\ \emptyset & \text{otherwise.} \end{cases}.$$

Also,

$$x \rightsquigarrow_t \epsilon = \begin{cases} a(x' \rightsquigarrow_{t'} \epsilon) & \text{if } e = i \\ \emptyset & \text{otherwise.} \end{cases}.$$

Further, $\epsilon \rightsquigarrow_t y = \epsilon$ if $t = y = \epsilon$. Otherwise, $\epsilon \rightsquigarrow_t y = \emptyset$.

Let $T \subseteq \{i, d\}^*$. Then

$$x \rightsquigarrow_T y = \bigcup_{t \in T} x \rightsquigarrow_t y$$

We extend this to languages as expected: Let $L_1, L_2 \subseteq \Sigma^*$ and $T \subseteq \{i, d\}^*$. Then

$$L_1 \rightsquigarrow_T L_2 = \bigcup_{x \in L_1} \bigcup_{y \in L_2} x \rightsquigarrow_T y.$$

Note that $\rightsquigarrow_T$ is not an associative operation on languages. Also, we note the difference of deletion on trajectories from the operation *splicing on routes* defined by Mateescu [15], which is a generalization of shuffle on trajectories which allows discarding symbols from either input word. In splicing on routes, the operation is always associative, and deletions may be made from either word without any co-ordination with the other word involved.

We consider the following examples of deletion along trajectories:

(a) if $T = i^*d^*$, then $\rightsquigarrow_T = /$, the right-quotient operation;

(b) if $T = d^*i^*$, then $\rightsquigarrow_T = \backslash$, the left-quotient operation;

(c) if $T = i^*d^*i^*$, then $\rightsquigarrow_T = \rightsquigarrow$, the deletion operation (see, e.g., Kari [9, 10]);

(d) if $T = (i + d)^*$, then $\rightsquigarrow_T$ is the scattered deletion operation (see, e.g., Ito *et al.* [8]);

(e) if $T = d^*i^*d^*$, then $\rightsquigarrow_T = \rightleftharpoons$, the bi-polar deletion operation (see, e.g., Kari [10]).

3 Closure and Characterization Results

Lemma 3.1 *If T,L_1, L_2 are regular, then $L_1 \rightsquigarrow_T L_2$ is also regular.*

Proof. The construction is straight-forward. Let M_1, M_2, M_T be DFAs for L_1, L_2, T, respectively, with

$$\begin{aligned} L_i &= (Q_i, \Sigma, \delta_i, q_i, F_i) \quad i = 1, 2 \\ T &= (Q_T, \{i, d\}, \delta_T, q_T, F_T) \end{aligned}$$

Then let $M = (Q_1 \times Q_2 \times Q_T, \Sigma, \delta, [q_1, q_2, q_T], F_1 \times F_2 \times F_T)$ be an NFA with δ given by

$$\delta([q_i, q_j, q_k], a) = \{[\delta_1(q_i, a), q_j, \delta_T(q_k, i)]\}$$

for all $[q_i, q_j, q_k] \in Q_1 \times Q_2 \times Q_T$ and $a \in \Sigma$. Further,

$$\delta([q_i, q_j, q_k], \epsilon) = \{[\delta_1(q_i, a), \delta(q_j, a), \delta_T(q_k, d)] \ : \ a \in \Sigma\}$$

for all $[q_i, q_j, q_k] \in Q_1 \times Q_2 \times Q_T$. We can verify that M accepts the proper language. ■

We now show that if one of L_1, L_2 or T is non-regular, then $L_1 \rightsquigarrow_T L_2$ may not be regular:

Theorem 3.2 *There exist languages L_1, L_2 and a set of trajectories T satisfying each of the following:*

(a) L_1 is a CFL, L_2 is a singleton and T is regular, but $L_1 \rightsquigarrow_T L_2$ in not regular;

(b) L_1, T are regular, and L_2 is a CFL, but $L_1 \rightsquigarrow_T L_2$ is not regular;

(c) L_1 is regular, L_2 is a singleton, and T is a CFL, but $L_1 \rightsquigarrow_T L_2$ is not regular.

In each case, the CFL may be chosen to be a linear CFL.

Proof. We first note the following identity:

$$L \rightsquigarrow_{i^*} \{\epsilon\} = L.$$

Thus, if we take any non-regular (linear) CFL L, we can establish (a).

For (b), we take the following languages:

$$\begin{aligned} L_1 &= a^*b^* \\ T &= (di)^* \\ L_2 &= \{a^n b^n \ : \ n \geq 0\} \end{aligned}$$

Note that L_2 is a non-regular (linear) CFL. With these languages, we get that $L_1 \rightsquigarrow_T L_2 = L_2\{\epsilon, b^2\}$, which is non-regular.

Finally, to establish part (c), we take

$$\begin{aligned} L_1 &= a^*\#b^* \\ T &= \{i^n d i^n \ : \ n \geq 0\} \\ L_2 &= \{\#\} \end{aligned}$$

We note that T is a non-regular linear CFL, and that

$$L_1 \rightsquigarrow_T L_2 = \{a^n b^n \ : \ n \geq 0\}.$$

This establishes the theorem. ■

In Section 4, we discuss non-regular sets of trajectories which preserve regularity. We have the following characterization of deletion along trajectories:

Theorem 3.3 *There exist morphisms $\rho_1, \rho_2, \tau, \varphi$ and a regular language R such that for all $L_1, L_2 \subseteq \Sigma^*$ and all $T \subseteq \{i, d\}^*$,*

$$L_1 \rightsquigarrow_T L_2 = \varphi\left(\rho_1^{-1}(L_1) \cap \rho_2^{-1}(L_2) \cap \tau^{-1}(T) \cap R\right).$$

Proof. Let $L_1, L_2 \subseteq \Sigma^*$ and $T \subseteq \{i, d\}^*$. Let $\hat{\Sigma} = \{\hat{a} \ : \ a \in \Sigma\}$ be a copy of Σ. Define the morphism $\rho_1 : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as follows:

$$\begin{aligned} \rho_1(\hat{a}) &= \rho_1(a) = a \quad \forall a \in \Sigma \\ \rho_1(i) &= \rho_1(d) = \epsilon \end{aligned}$$

Define $\rho_2 : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as follows:

$$\begin{aligned} \rho_2(\hat{a}) &= a \quad \forall a \in \Sigma \\ \rho_2(a) &= \epsilon \quad \forall a \in \Sigma \\ \rho_2(d) &= \rho_2(i) = \epsilon \end{aligned}$$

Define $\tau : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as follows

$$\begin{aligned} \tau(\hat{a}) &= \tau(a) = \epsilon \quad \forall a \in \Sigma \\ \tau(i) &= i \\ \tau(d) &= d \end{aligned}$$

We define $\varphi : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as

$$\begin{aligned} \varphi(\hat{a}) &= \epsilon \quad \forall a \in \Sigma \\ \varphi(a) &= a \quad \forall a \in \Sigma \\ \varphi(i) &= \varphi(d) = \epsilon \end{aligned}$$

Finally, we note that the result follows on letting $R = (i\Sigma + d\hat{\Sigma})^*$. ■

Recall that a cone (or full trio) is a class of languages closed under morphism, inverse morphism and intersection with regular languages [17, Sect. 3]. Thus, we have the following corollary:

Corollary 3.4 *Let $\mathcal{L}$ be a cone. Then let L_1, L_2, T be languages such that two are regular and the third is in $\mathcal{L}$. Then $L_1 \rightsquigarrow_T L_2 \in \mathcal{L}$.*

Note that the closure of cones under quotient with regular sets [7, Thm. 11.3] is a specific instance of Corollary 3.4. We also note that the CFLs are a cone, thus we have the following corollary (a direct construction is also possible):

Corollary 3.5 *Let T, L_1, L_2 be languages such that one is a CFL and the other two are regular languages. Then $L_1 \rightsquigarrow_T L_2$ is a CFL.*

Lemma 3.1 can also be proved by appealing to Theorem 3.3. The following result shows that even if we shuffle CFLs along a regular trajectory, the result may not be a CFL:

Theorem 3.6 *The CFLs are not closed under deletion along regular sets of trajectories.*

Proof. The result is immediate, since is is known (see, Ginsburg and Spanier [5, Thm. 3.4]) that the CFLs are not closed under right quotient (given by the trajectory $T = i^*d^*$). ■

We also have the following result:

Theorem 3.7 *There exist $L_1, L_2 \subseteq \Sigma^*$, $T \subseteq \{i, d\}^*$, such that L_1, T are CFLs, L_2 is a singleton, but $L_1 \rightsquigarrow_T L_2$ is not a CFL.*

Proof. Let $\Sigma = \{a, b, c, \#\}$. Then let

$$\begin{aligned} L_1 &= \{a^n b^n \# c^m \ : \ n, m \geq 0\}; \\ L_2 &= \{\#\}; \\ T &= \{i^{2n} d i^n \ : \ n \geq 0\}. \end{aligned}$$

Note that L_1, T are indeed CFLs. Then we can verify that

$$L_1 \rightsquigarrow_T L_2 = \{a^n b^n c^n \ : \ n \geq 0\},$$

which is not a CFL. ■

We have one final case to deal with:

Theorem 3.8 *There exist $L_1, L_2 \subseteq \Sigma^*$, $T \subseteq \{i, d\}^*$, such that L_2, T are CFLs, L_1 is regular, but $L_1 \rightsquigarrow_T L_2$ is not a CFL.*

Proof. Let $\Sigma = \{a, b, c\}$. Then let

$$\begin{aligned} L_1 &= a^* b^* c^+; \\ L_2 &= \{a^n b^n c \ : \ n \geq 0\}; \\ T &= \{(di)^{2n} d i^n \ : \ n \geq 0\}. \end{aligned}$$

Then we can verify that $L_1 \rightsquigarrow_T L_2$ is the non-CF language

$$\{a^n b^n c^n : n \geq 0\} \cup \{a^n b^{n+1} c^{n+2} : n \geq 0\} \cup \{a^n b^{n+2} c^{n+1} : n \geq 0\} \cup \{a^{n+1} b^n c^{n+2} : n \geq 0\}.$$

This completes the proof. ■

Note that the context-sensitive languages (CSLs) are not a cone, since they are not closed under arbitrary morphism. Thus, Corollary 3.4 does not apply to the CSLs. We now construct an example demonstrating non-closure of the CSLs under deletion of a regular language along a regular set of trajectories.

This construction is similar to one used by Daley and Kari [2, Prop. 2.4]. We will require the following theorem (see Salomaa [22]):

Theorem 3.9 *Let Σ be a language and $a, b \notin \Sigma$. For all recursively enumerable languages $L \subseteq \Sigma^*$, there exists a CSL $L_1 \subseteq a^*bL$ such that for all $x \in L$, there exists some $a^ibx \in L_1$.*

Theorem 3.10 *There exist a CSL L, a regular set of trajectories $T \subseteq \{i, d\}^*$ and a regular language R such that $L \rightsquigarrow_T R$ is not a CSL.*

Proof. Let L be a recursively enumerable non-CS language. Let L_1 be the CSL given by Theorem 3.9.

Consider $T = d^*i^*$ (i.e., left quotient) and $R = a^*b$. Then $L_1 \rightsquigarrow_T R = L$. This establishes the result. ■

4 Regularity-Preserving Non-Regular Trajectories

Consider the following result of Mateescu *et al.* [16, Thm. 5.1]: if $L_1 \sqcup\!\sqcup_T L_2$ is regular for all regular languages L_1, L_2, then T is regular. This result is clear upon noting that for all T, $0^* \sqcup\!\sqcup_T 1^* = T$.

However, in this section, we note that the same result does not hold if we replace "shuffle on trajectories" by "deletion along trajectories". In particular, we demonstrate a class of non-regular sets of trajectories $\mathcal{C}$ such that for all regular languages L_1, L_2, and for all $H \in \mathcal{C}$, $L_1 \rightsquigarrow_H L_2$ is regular. We also characterize all $H \subseteq i^*d^*$ which preserve regularity, and give some examples of non-CF trajectories which preserve regularity.

As motivation, we begin with a basic example. Let Σ be an alphabet. Let $H = \{i^nd^n \ : \ n \geq 0\}$. Note that

$$L_1 \rightsquigarrow_H L_2 = \{x \in \Sigma^* \ : \ \exists y \in L_2 \text{ such that } xy \in L_1 \text{ and } |x| = |y|\}.$$

We can establish directly (by constructing an NFA) that for all regular languages $L_1, L_2 \subseteq \Sigma^*$, the language $L_1 \rightsquigarrow_H L_2$ is regular. However, H is a non-regular CFL.

Remark that $L_1 \rightsquigarrow_H L_2$ is similar to proportional removals studied by Stearns and Hartmanis [24], Seiferas and McNaughton [23], Kosaraju [13, 12], Kozen [14], Zhang [26], the author [3] and others. In particular, we note the case of $\frac{1}{2}(L)$, given by

$$\frac{1}{2}(L) = \{x \in \Sigma^* : \exists y \in \Sigma^* \text{ such that } xy \in L \text{ and } |x| = |y|\}.$$

The operation $\frac{1}{2}(L)$ is one of a class of operations which preserve regularity. Seiferas and McNaughton completely characterize those binary relations $r \subseteq \mathbb{N}^2$ such that the operation

$$P(L, r) = \{x \in \Sigma^* : \exists y \in \Sigma^* \text{ such that } xy \in L \text{ and } r(|x|, |y|)\}$$

preserves regularity (We note that $\frac{1}{2}(L)$ can be simulated by splicing on the CF route $\{1^n\overline{1}^n\overline{2}^m \ : \ n, m \geq 0\}$; see Mateescu [15] for details).

Recall that a set A is ultimately periodic (or simply u.p.) if there exists n_0, p such that $\forall n \geq n_0 (n \in A \iff n + p \in A)$. A relation r is u.p.-preserving if A u.p. implies

$$r^{-1}(A) = \{i \ : \ \exists j \in A \text{ such that } r(i, j)\}$$

is also u.p. Then, the r that preserve regularity are precisely the u.p.-preserving relations.

We note the inclusion

$$L_1 \rightsquigarrow_H L_2 \subseteq \frac{1}{2}(L_1) \cap L_1/L_2.$$

However, equality does not hold in general. Consider the languages

$$\begin{aligned} L_1 &= \{0^2, 0^4\} \\ L_2 &= \{0^3\} \end{aligned}$$

Then note that $0 \in \frac{1}{2}(L_1) \cap L_1/L_2$. However, $0 \notin L_1 \rightsquigarrow_H L_2$. Thus, we note that

$$L_1 \rightsquigarrow_H L_2 \neq \frac{1}{2}(L_1) \cap L_1/L_2$$

in general.

We now consider arbitrary relations $r \subseteq \mathbb{N}^2$ for which

$$H_r = \{i^n d^m \ : \ r(n,m)\} \subseteq i^* d^*$$

preserves regularity. By modifying the construction of Seiferas and McNaughton, we obtain the following result:

Theorem 4.1 *The operation $\rightsquigarrow_{H_r}$ is regularity-preserving iff r is u.p.-preserving.*

Proof. Assume that $\rightsquigarrow_{H_r}$ is preserves regularity. Then $L \rightsquigarrow_{H_r} \Sigma^*$ is regular for all regular languages L. But

$$L \rightsquigarrow_{H_r} \Sigma^* = P(L, r).$$

Thus, r must be u.p.-preserving.

For the reverse implication, we modify the construction of Seiferas and McNaughton [23, Thm. 1]. Let M_1 be the minimal complete DFA for L_1: $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$. Then, for all $q \in Q_1$, we let $L_1^{(q)}$ be the language recognized by the DFA $M_1^{(q)} = (Q_1, \Sigma, \delta_1, q_0, \{q\})$. Let R_q be the language recognized by the DFA $N_1^{(q)} = (Q_1, \Sigma, \delta_1, q, F)$.

As M_1 is complete, $\Sigma^* = \bigcup_{q \in Q} L_1^{(q)}$. Thus,

$$L_1 \rightsquigarrow_{H_r} L_2 = \bigcup_{q \in Q} (L_1 \rightsquigarrow_{H_r} L_2) \cap L_1^{(q)}.$$

Thus, it suffices to demonstrate that $(L_1 \rightsquigarrow_{H_r} L_2) \cap L_1^{(q)}$ is regular. But we now note that

$$\begin{aligned} (L_1 \rightsquigarrow_{H_r} L_2) \cap L_1^{(q)} &= \{x \in L_1^{(q)} \ : \ \exists y \in L_2 \text{ such that } xy \in L_1 \text{ and } r(|x|, |y|)\} \\ &= \{x \in L_1^{(q)} \ : \ \exists y \in (R_q \cap L_2) \text{ such that } r(|x|, |y|)\} \\ &= \{x \in \Sigma^* \ : \ \exists y \in (R_q \cap L_2) \text{ such that } r(|x|, |y|)\} \cap L_1^{(q)} \\ &= \{x \in \Sigma^* \ : \ |x| \in r^{-1}(\{|y| : y \in (R_q \cap L_2)\})\} \cap L_1^{(q)} \end{aligned}$$

Note that if L is regular, $\{|y| \ : \ y \in L\}$ is a u.p. set. As r is u.p.-preserving, $r^{-1}(\{|y| \ : \ y \in R_q \cap L_2)\})$ is also u.p. Further, it is an easy exercise to construct a DFA for the language $\{x \in \Sigma^* \ : \ |x| \in A\}$ for any u.p. set A. ■

Note that in general, the equality

$$L_1 \rightsquigarrow_{H_r} L_2 = P(L_1, r) \cap L_1/L_2$$

does not hold. We note some particular examples of regularity-preserving trajectories:

(a) Consider the relation $e = \{(n, 2^n) \ : \ n \geq 0\}$. Then H_e preserves regularity (see, e.g., Zhang [26, Sect. 3]). However, H_e is not CF. The set H_e is, however, a linear conjunctive language (see Okhotin [20] for the definition of conjunctive and linear conjunctive languages, and for the proof that H_e is linear conjunctive).

(b) Consider the relation $f = \{(n, n!) \ : \ n \geq 0\}$. Then H_f preserves regularity (see again Zhang [26, Thm. 5.1]). However, H_f is not a CFL, nor a linear conjunctive language [20].

Thus, we note that there are non-CF trajectories which preserve regularity.

5 Deletion as an Inverse of Shuffle on Trajectories

Given two binary word operations $\diamond, \star : (\Sigma^*)^2 \to \Sigma^*$, we say that $\diamond$ is a left-inverse of $\star$ [10, Defn. 4.1] if, for all $u, v, w \in \Sigma^*$,

$$w \in u \star v \iff u \in w \diamond v.$$

Then we have the following characterization of left-inverses:

Theorem 5.1 *Let τ be the morphism which maps $0 \to i$ and $1 \to d$. Let $T \subseteq \{0,1\}^*$ be a set of trajectories. Then $\sqcup\!\sqcup_T$ and $\rightsquigarrow_{\tau(T)}$ are left-inverses of each other.*

Proof. We show that for all $t \in \{0,1\}^*$, $(w \in u \sqcup\!\sqcup_t v) \iff (u \in w \rightsquigarrow_{\tau(t)} v)$. The proof is by induction on $|w|$. For $|w| = 0$, we have $w = \epsilon$. Thus, by definition of $\sqcup\!\sqcup_t$ and $\rightsquigarrow_t$, we have that

$$\begin{aligned} & \epsilon \in u \sqcup\!\sqcup_t v \\ \iff \quad & u = v = t = \epsilon \\ \iff \quad & u \in (\epsilon \rightsquigarrow_{\tau(t)} v) \end{aligned}$$

Now, assume that the result is true for all strings shorter than w. Let $w = aw'$ for $a \in \Sigma$. First, assume that $aw' \in u \sqcup\!\sqcup_t v$. As $|t| = |w|$, we have that $t \neq \epsilon$. Let $t = et'$ for some $e \in \{0,1\}$ and let u', v' be strings such that

$$w' \in u' \sqcup\!\sqcup_{t'} v';$$

such strings necessarily exist by definition of $\sqcup\!\sqcup_T$. Thus, by induction,

$$u' \in w' \rightsquigarrow_{\tau(t')} v'.$$

Let $e' = \tau(e)$. We have two cases:

(a) if $e = 0$, then $v = v'$ and $u = au'$. Thus $e' = i$ and

$$\begin{aligned} (w \rightsquigarrow_{\tau(t)} v) \ &= \ (aw' \rightsquigarrow_{e'\tau(t')} v') \\ &= \ a(w' \rightsquigarrow_{\tau(t')} v') \\ &\ni \ au' = u. \end{aligned}$$

(b) If $e = 1$, then $u = u'$ and $v = av'$. Thus $e' = d$ and

$$\begin{aligned}(w \rightsquigarrow_{\tau(t)} v) &= (aw' \rightsquigarrow_{e'\tau(t')} av') \\ &= (w' \rightsquigarrow_{\tau(t')} v') \\ &\ni u' = u.\end{aligned}$$

Thus, we have that in both cases $u \in w \rightsquigarrow_{\tau(t)} v$.

Now, let us assume that $u \in w \rightsquigarrow_{\tau(t)} v$. Then as $|t| = |\tau(t)| = |w| \geq 1$, let $t = et'$ for some $e \in \{0, 1\}$ and let u', v' be the strings such that,

$$u' \in (w' \rightsquigarrow_{\tau t'} v').$$

By induction,

$$w' \in u' \sqcup\!\sqcup_t v'.$$

We again have two cases:

(a) if $e = 0$, then $\tau(e) = i$. Then necessarily $u = au'$, $v' = v$. Thus

$$\begin{aligned}(u \sqcup\!\sqcup_t v) &= (au' \sqcup\!\sqcup_{et'} v') \\ &= a(u' \sqcup\!\sqcup_{t'} v') \\ &\ni aw' = w.\end{aligned}$$

(b) if $e = 1$, Then $\tau(e) = i$. Then we have that $u = u'$ and $v = av'$. Thus

$$\begin{aligned}(u \sqcup\!\sqcup_t v) &= (u' \sqcup\!\sqcup_{et'} av') \\ &= a(u' \sqcup\!\sqcup_{t'} v') \\ &\ni aw' = w.\end{aligned}$$

Thus $w \in u \sqcup\!\sqcup_t v$. This completes the proof. ■

We note that Theorem 5.1 agrees with the observations of Kari [10, Obs. 4.7].

5.1 Solving $X \sqcup\!\sqcup_T L = R$ and $X \rightsquigarrow_T L = R$

The following is a result of Kari [10, Thm. 4.6]:

Theorem 5.2 *Let L, R be languages over Σ and $\diamond, \star$ be two binary word operations, which are left-inverses to each other. If the equation $X \diamond L = R$ has a solution $X \subseteq \Sigma^*$, then the language*

$$R' = \overline{\overline{R} \star L}$$

is also a solution of the equation. Moreover, R' is a superset of all other solutions of the equation.

By Theorem 5.2, Theorem 5.1 and Lemma 3.1, we note the following corollary:

Corollary 5.3 *Let $T \subseteq \{0,1\}^*$. Let T, L, R be regular languages. Then it is decidable whether the equation*

$$X \sqcup\!\sqcup_T L = R$$

has a solution X.

The idea is the same as discussed by Kari [10, Thm. 2.3]: we compute R' given in Theorem 5.2, and check whether R' is a solution to the desired equation. Since all languages involved are regular, the constructions are effective and we can test for equality of regular languages. Also, we note the following corollary, which is established in the same manner as Corollary 5.3:

Corollary 5.4 *Let $T \subseteq \{i,d\}^*$. Let T, L, R be regular languages. Then it is decidable whether the equation*

$$X \rightsquigarrow_T L = R$$

has a solution X.

5.2 Solving $L \sqcup\!\sqcup_T X = R$

Given two binary word operations $\diamond, \star : (\Sigma^*)^2 \to \Sigma^*$, we say that $\diamond$ is a right-inverse of $\star$ if, for all $u, v, w \in \Sigma^*$,

$$w \in u \star v \iff v \in u \diamond w.$$

Let $\diamond$ be a binary word operation. The word operation $\diamond^r$ given by $u \diamond^r v = v \diamond u$ is called *reversed* $\diamond$ [10].

We can repeat the above arguments for right-inverses instead of left-inverses. In all cases, the proofs are similar to those of the previous section. Thus, we simply state the results:

Theorem 5.5 *Let π be the morphism which maps $0 \to d$ and $1 \to i$. Let $T \subseteq \{0,1\}^*$ be a set of trajectories. Then $\sqcup\!\sqcup_T$ and $(\rightsquigarrow_{\pi(T)})^r$ are right-inverses of each other.*

This again agrees with the observations of Kari [10, Obs. 4.4].

Corollary 5.6 *Let $T \subseteq \{0,1\}^*$. Let T, L, R be regular languages. Then it is decidable whether the equation*

$$L \sqcup\!\sqcup_T X = R$$

has a solution X.

We note that Câmpeanu *et al.* have recently investigated the decidability of the existence of solutions to the equation $X_1 \sqcup\!\sqcup X_2 = R$ (i.e., unrestricted shuffle given by $T = (0+1)^*$) where X_1, X_2 are unknown and R is regular [1].

5.3 Solving $\{x\} \sqcup_T L = R$

In this section, we briefly address the problem of finding solutions to equations of the form

$$\{x\} \sqcup_T L = R$$

where T is a fixed regular set of trajectories, L, R are regular languages, and x is an unknown word. This is a generalization of the results of Kari [10].

Theorem 5.7 *Let Σ be an alphabet. Let $T \subseteq \{0,1\}^*$ be a fixed regular set of trajectories. Then for all regular languages $R, L \subseteq \Sigma^*$, it is decidable whether there exists a word $x \in \Sigma^*$ such that $\{x\} \sqcup_T L = R$.*

Proof. Let $r = \min\{|y| \ : \ y \in R\}$. Then note that $|z| = |x| + |y|$ for all $z \in x \sqcup_T y$ (regardless of T). Thus, it is clear that if x exists satisfying $\{x\} \sqcup_T L = R$, then $|x| \leq r$. Our algorithm then simply considers all strings x of length at most r, and checks whether $\{x\} \sqcup_T L = R$ holds. ■

5.4 Solving $L \rightsquigarrow_T X = R$

We now consider the decidability of solutions to the equation $L \rightsquigarrow_T X = R$ where T is a fixed set of trajectories, L, R are regular languages and X is unknown.

This involves considering the right-inverse of $\rightsquigarrow_T$ for all $T \subseteq \{i,d\}^*$. However, unlike the left-inverse of $\rightsquigarrow_T$, the right-inverse of $\rightsquigarrow_T$ is again a deletion operation. Let $\overline{\cdot} : \{i,d\}^* \to \{i,d\}^*$ be the morphism given by $\overline{i} = d$ and $\overline{d} = i$.

Theorem 5.8 *Let $T \subseteq \{i,d\}^*$ be a set of trajectories. The operation $\rightsquigarrow_T$ has right-inverse $\rightsquigarrow_{\overline{T}}$.*

Proof. Let Σ be an arbitrary alphabet. We establish that for all $x, y, z \in \Sigma^*$, and all $t \in \{i,d\}^*$,

$$(x \in y \rightsquigarrow_t z) \iff (z \in y \rightsquigarrow_{\overline{t}} x).$$

The proof is by induction on $|y|$.

For $|y| = 0$, $y = \epsilon$. Thus

$$\begin{aligned} & x \in \epsilon \rightsquigarrow_t z \\ \iff \quad & x = t = z = \epsilon \\ \iff \quad & z \in y \rightsquigarrow_{\overline{t}} x. \end{aligned}$$

Assume the result holds for all strings with length less than $|y|$. Consider $y = ay'$ for $a \in \Sigma$. Assume that $x \in ay' \rightsquigarrow_t z$. Then as $|t| = |y|$, $t = et'$ for some $e \in \{i,d\}$. We have two cases:

(a) if $e = i$, then $x \in a(y' \rightsquigarrow_{t'} z)$. Assume $x = ax'$ for some $x' \in \Sigma^*$. Then $x' \in (y' \rightsquigarrow_{t'} z)$ and by induction $z \in (y' \rightsquigarrow_{\overline{t'}} x')$. Consider then that

$$\begin{aligned} y \rightsquigarrow_{\overline{t}} x \ &= \ ay \rightsquigarrow_{d\overline{t'}} ax' \\ &= \ y' \rightsquigarrow_{\overline{t'}} x' \\ &\ni \ z. \end{aligned}$$

Thus $z \in \rightsquigarrow_{\overline{t}} x$, as required.

(b) if $e = d$, then $z = az'$ for some $z' \in \Sigma^*$ and $x \in y' \rightsquigarrow_{t'} z'$. As $|y'| < |y|$, by induction, $z' \in y' \rightsquigarrow_{\overline{t'}} x$. Thus

$$\begin{aligned} y \rightsquigarrow_{\overline{t}} x &= ay \rightsquigarrow_{i\overline{t'}} x \\ &= a(y \rightsquigarrow_{\overline{t'}} x) \\ &\ni az' = z. \end{aligned}$$

Thus, we have established that

$$x \in (y \rightsquigarrow_t z) \Rightarrow z \in (y \rightsquigarrow_{\overline{t}} x).$$

The reverse implication follows on noting that $\overline{\overline{t}} = t$. ■

We note that Theorem 5.8 agrees with the observations of Kari [10, Obs. 4.4]. Also, we note the following result:

Corollary 5.9 *Let $T \subseteq \{i, d\}^*$. Let T, L, R be regular languages. Then it is decidable whether the equation*

$$L \rightsquigarrow_T X = R$$

has a solution X.

5.5 Solving $\{x\} \rightsquigarrow_T L = R$

In this section, we are concerned with decidability of the existence of solutions to the equation

$$\{x\} \rightsquigarrow_T L = R$$

where x is a string in Σ^*, and L, R, T are regular languages. Equations of this form have previously been considered by Kari [10]. Our constructions generalize those of Kari directly.

Let τ again be the renaming $0 \to i$, $1 \to d$. We begin with the following lemma:

Lemma 5.10 *Let Σ be an alphabet. Then for all sets of trajectories $T \subseteq \{i, d\}^*$, the following equality holds:*

$$\overline{(\overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L)} = \{x \ : \ \{x\} \rightsquigarrow_T L \subseteq R\}.$$

Proof. Let x be a string such that $\{x\} \rightsquigarrow_T L \subseteq R$, and assume, contrary to what we want to prove, that $x \in \overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L$. Then there exists $y \in \overline{R}, z \in L$ and $t \in \tau^{-1}(T)$ such that $x \in y \sqcup\!\sqcup_t z$. By Theorem 5.1,

$$y \in x \rightsquigarrow_{\tau(t)} z.$$

As $\tau(t) \in T$, we conclude that $y \in \overline{(\{x\} \rightsquigarrow_T L)} \cap \overline{R}$. Thus $\{x\} \rightsquigarrow_T L \subseteq R$ does not hold, contrary to our choice of x. Thus $x \in \overline{(\overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L)}$.

For the reverse inclusion, let $x \in \overline{(\overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L)}$. Further, assume that $(\{x\} \rightsquigarrow_T L) \cap \overline{R} \neq \emptyset$. In particular, there exist strings $z \in L$ and $t \in T$ such that

$$x \rightsquigarrow_t z \cap \overline{R} \neq \emptyset.$$

Let y be some string in this intersection. As $y \in x \rightsquigarrow_t z$, by Theorem 5.1, we have that $x \in y \sqcup\!\sqcup_{\tau^{-1}(t)} z$. Thus, $x \in \overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L$, contrary to our choice of x. This proves the result. ■

Thus, we can state the main result of this section:

Theorem 5.11 *Let Σ be an alphabet. Let $T \subseteq \{i, d\}^*$ be an arbitrary regular set of trajectories. Then the problem "Does there exist a word x such that $\{x\} \rightsquigarrow_T L = R$ is decidable for regular languages L, R.*

Proof. Let L, R be regular languages. We note that if R is infinite, then the answer to our problem is no; there can only be finitely many deletions along trajectory T from a finite string x. Thus, assume that R is finite. Then we can construct the following regular language:

$$P = \overline{(\overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L)} - \bigcup_{S \subsetneq R} \overline{(\overline{S} \sqcup\!\sqcup_{\tau^{-1}(T)} L)}.$$

Note that $\subsetneq$ denotes proper inclusion. We claim that $P = \{x \ : \ \{x\} \rightsquigarrow_T L = R\}$.

Assume $x \in P$. Then by Lemma 5.10, we have that

$$x \ \in \ \{x \ : \ \{x\} \rightsquigarrow_T L \subseteq R\} \tag{5.1}$$
$$x \ \notin \ \{x \ : \ \{x\} \rightsquigarrow_T L \subseteq S \subsetneq R\} \tag{5.2}$$

Thus, we must have that $\{x\} \rightsquigarrow_T L = R$, since $\{x\} \rightsquigarrow_T L$ is a subset of R, but is not contained in any proper subset of R.

Similarly, if $\{x\} \rightsquigarrow_T L = R$, then by Lemma 5.10, we have that $x \in \overline{(\overline{R} \sqcup\!\sqcup_{\tau^{-1}(T)} L)}$. But as $\{x\} \rightsquigarrow_T L$ is not contained in any S with $S \subsetneq R$, we have that $x \notin \bigcup_{S \subsetneq R} \overline{(\overline{S} \sqcup\!\sqcup_{\tau^{-1}(T)} L)}$. Thus, $x \in P$.

Thus, if R is finite, to decide if a string x exists satisfying $\{x\} \rightsquigarrow_T L = R$, we construct P and test if $P \neq \emptyset$. Since P will be regular, this can be done effectively (as we have noted, if R is infinite, we answer no). ■

6 Recognizing Deletion Along Trajectories

We now consider the problem of giving a monoid recognizing deletion along trajectories.

For a background on recognition of formal languages by monoids, please consult Pin [21]. A monoid is a semigroup with unit element. Let $L \subseteq \Sigma^*$ be a language. We say that a monoid M recognizes L if there exists a morphism $\varphi : \Sigma^* \to M$ and a subset $F \subseteq M$ such that $L = \varphi^{-1}(F)$.

The following is a characterization of the regular languages due to Kleene (see, e.g., Pin [21, p. 17]):

Theorem 6.1 *A language is regular iff it is recognized by a finite monoid.*

Consider arbitrary regular languages $L_1, L_2 \subseteq \Sigma^*$ and $T \subseteq \{i, d\}^*$. Then our goal is to construct a monoid recognizing $L_1 \rightsquigarrow_T L_2$.

Let M_1, M_2, M_T be finite monoids recognizing L_1, L_2, L_T, with morphisms $\varphi_i : \Sigma^* \to M_i$ for $i = 1, 2$, $\varphi_T : \{i, d\} \to M_T$ and subsets F_1, F_2, F_T, respectively.

As in Harju *et al.* [6], we consider the monoid $\mathcal{P}(M_1 \times M_2 \times M_T)$ consisting of all subsets of $M_1 \times M_2 \times M_T$. The monoid operation is given by

$$AB = \{xy \ : x \in A, y \in B\}$$

for all $A, B \in \mathcal{P}(M_1 \times M_2 \times M_T)$.

We can now establish that $\mathcal{P}(M_1 \times M_2 \times M_T)$ recognizes $L_1 \rightsquigarrow_T L_2$. We first define a subset $D \subseteq M_1 \times M_2 \times M_T$ which will be useful:

$$D = \{[\varphi_1(x), \varphi_2(x), \varphi_T(d^{|x|})] \ : \ x \in \Sigma^*\}.$$

Then we define $\varphi : \Sigma^* \to \mathcal{P}(M_1 \times M_2 \times M_T)$ by giving its action on each element $a \in \Sigma$:

$$\varphi(a) = \{[\varphi_1(xa), \varphi_2(x), \varphi_T(d^{|x|}i)] \ : \ x \in \Sigma^*\}.$$

Then, we note that for all $y \in \Sigma^*$,

$$\varphi(y)D = \{[\varphi_1(\alpha), \varphi_2(\beta), \varphi_T(t)] \ : \ y \in \alpha \rightsquigarrow_t \beta \text{ where } \alpha, \beta \in \Sigma^*, t \in \{i, d\}^*\} \qquad (6.3)$$

Thus, it suffices to take

$$F = \{K \in \mathcal{P}(M_1 \times M_2 \times M_T) \ : \ KD \cap (F_1 \times F_2 \times F_T) \neq \emptyset\}.$$

Thus, considering (6.3), we have that

$$L_1 \rightsquigarrow_T L_2 = \varphi^{-1}(F).$$

This establishes that $\mathcal{P}(M_1 \times M_2 \times M_T)$ recognizes $L_1 \rightsquigarrow_T L_2$.

7 Conclusion

We have defined deletion along trajectories, and examined its closure properties. Deletion along trajectories is shown to be a useful generalization of many deletion-based operations which have been studied in the literature. The closure properties of differ from that of shuffle on trajectories in that there exist non-regular and non-CF sets of trajectories which define operations which preserve regularity.

We have also demonstrated that deletion along trajectories constitutes an elegant inverse to shuffle along trajectories operations. This leads to positive decidability results for equations involving shuffle on trajectories and deletion along trajectories.

References

[1] CÂMPEANU, C., SALOMAA, K., AND VÁGVÖLGYI, S. Shuffle decompositions of regular languages. *Int. J. Found. Comp. Sci. 13*, 6 (2002), 799–816.

[2] DALEY, M., AND KARI, L. Some properties of ciliate bio-operations. In *DLT 2002: Developments in Language Theory, Sixth International Conference* (2002), pp. 122–139.

[3] DOMARATZKI, M. State complexity of proportional removals. *To appear, J. Automata, Languages and Combinatorics 7*, 4 (2002), 455–468.

[4] DOMARATZKI, M., AND SALOMAA, K. State complexity of shuffle on trajectories. In *Descriptional Complexity of Formal Systems (DCFS)* (2002), pp. 81–94.

[5] GINSBURG, S., AND SPANIER, E. Quotients of context-free languages. *J. ACM 10*, 4 (1963), 487–492.

[6] HARJU, T., MATEESCU, A., AND SALOMAA, A. Shuffle on trajectories: The Schützenberger product and related operations. In *MFCS 1998*, no. 1450 in LNCS. Springer-Verlag, 1998, pp. 503–511.

[7] HOPCROFT, J. E., AND ULLMAN, J. D. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[8] ITO, M., KARI, L., AND THIERRIN, G. Shuffle and scattered deletion closure of languages. *Theor. Comp. Sci. 245* (2000), 115–133.

[9] KARI, L. Generalized derivatives. *Fund. Inf. 18* (1993), 27–39.

[10] KARI, L. On language equations with invertible operations. *Theor. Comp. Sci. 132* (1994), 129–150.

[11] KARI, L., AND THIERRIN, G. Maximal and minimal solutions to language equations. *J. Comp. Sys. Sci. 53* (1996), 487–496.

[12] KOSARAJU, S. Correction to "Regularity Preserving Functions". *ACM SIGACT News 6*, 3 (1974), 22.

[13] KOSARAJU, S. Regularity preserving functions. *ACM SIGACT News 6*, 2 (1974), 16–17.

[14] KOZEN, D. On regularity-preserving functions. Tech. Rep. TR95-1559, Department of Computer Science, Cornell University, 1995.

[15] MATEESCU, A. Splicing on routes: a framework of DNA computation. In *Unconvential Models of Computation* (1998), C. Calude, J. Casti, and M. Dinneen, Eds., Springer, pp. 273–285.

[16] MATEESCU, A., ROZENBERG, G., AND SALOMAA, A. Shuffle on trajectories: Syntactic constraints. *Theor. Comp. Sci. 197* (1998), 1–56.

[17] MATEESCU, A., AND SALOMAA, A. Aspects of classical language theory. In *Handbook of Formal Languages, Vol. I*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, 1997, pp. 175–246.

[18] MATEESCU, A., AND SALOMAA, A. Nondeterministic trajectories. In *Formal and Natural Computing*, vol. 2300 of *LNCS*. Springer-Verlag, 2002, pp. 96–106.

[19] MATEESCU, A., SALOMAA, K., AND YU, S. On fairness of many-dimensional trajectories. *J. Automata, Languages and Combinatorics 5* (2000), 145–157.

[20] OKHOTIN, A. Automaton representation of linear conjunctive languages. In *DLT 2002: Developments in Language Theory, Sixth International Conference* (2002), pp. 327–341.

[21] PIN, J.-E. *Varities of Formal Languages*. Plenum, 1986.

[22] SALOMAA, A. *Formal Languages*. Academic Press, 1973.

[23] SEIFERAS, J., AND MCNAUGHTON, R. Regularity-preserving relations. *Theoretical Computer Science 2* (1976), 147–154.

[24] STEARNS, R., AND HARTMANIS, J. Regularity preserving modifications of regular expressions. *Inf. and Cont. 6*, 1 (1963), 55–69.

[25] YU, S. Regular languages. In *Handbook of Formal Languages, Vol. I*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, 1997, pp. 41–110.

[26] ZHANG, G.-Q. Automata, boolean matrices, and ultimate periodicity. *Information and Computation 152* (1999), 138–154.