Deletion along Trajectories

Michael Domaratzki [1]

School of Computing, Queen's University, Kingston, Ontario, Canada K7L 3N6.

Abstract

We describe a new way to model deletion operations on formal languages, called *deletion along trajectories*. We examine its closure properties, which differ from those of shuffle on trajectories, previously introduced by Mateescu *et al.* In particular, we define classes of non-regular sets of trajectories such that the associated deletion operation preserves regularity. Our results give uniform proofs of closure properties of the regular languages for several deletion operations.

We also show that deletion along trajectories serves as an inverse to shuffle on trajectories. This leads to results on the decidability of certain language equations, including those of the form $L \sqcup\!\sqcup_T X = R$, where L, R are regular languages and X is unknown.

Key words: deletion along trajectories, shuffle on trajectories, language equations, regular languages

1 Introduction

Shuffle on trajectories, defined by Mateescu *et al.* [19], unifies operations which insert all the symbols of one word into another (see Section 2 for definitions). Operations in the literature generalized by shuffle on trajectories include concatenation, reverse and bi-concatenation, arbitrary, literal and perfect shuffles, and others. This formalism has proven to be very powerful, and much work has recently been done on shuffle on trajectories [6,8,21,22]. Mateescu has defined an extension of shuffle on trajectories called splicing on routes, which generalizes operations on DNA strands [18].

Concurrent to this research, Kari and others [12,14] have done research into the inverses of insertion- and shuffle-like operations, which have yielded decid-

Email address: domaratz@cs.queensu.ca (Michael Domaratzki).

[1] Research supported by an NSERC PGS-B graduate scholarship.

ability 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-like operations such as *deletion*, *quotient*, *scattered deletion* and *bipolar deletion* [12].

In this paper, we introduce the notion of *deletion along trajectories*, which is the equivalent of shuffle on trajectories for deletion-like operations. We show how it unifies operations such as deletion, quotient, scattered deletion and others. We investigate the closure properties of deletion along trajectories. 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 $L \sqcup\!\sqcup_T X = R$ for regular languages L and R have a solution X, for any regular set T of trajectories.

We also investigate those T which are not regular but for which the deletion along the set of trajectories T preserves regularity. Theorems 4.1 and 4.2 explicitly define classes of sets of trajectories, which include non-regular sets, which preserve regularity. These theorems give uniform proofs of certain closure properties for the regular languages.

2 Definitions

For additional background in formal languages and automata theory, see Yu [27] or Hopcroft and Ullman [9]. 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. If $L_1, \ldots, L_k \subseteq \Sigma^*$ are languages, we use the notation $\prod_{i=1}^{k} L_i = L_1 L_2 \cdots L_k$. If L is a language and k is a natural number, then we denote $L^{\leq k} = \{u_1 u_2 \cdots u_i \ : \ i \leq k, u_j \in L \ \forall 1 \leq j \leq i\}$.

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, $\delta : Q \times \Sigma \to Q$ is a transition function, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of 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 accepted* by M, denoted $L(M)$, is the set of all words accepted 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 as in the deterministic case, while the nondeterministic transition function is given by $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$. 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.

We denote by $\mathbb{N}$ the set of non-negative integers: $\mathbb{N} = \{0, 1, 2, \dots\}$. Let $I \subseteq \mathbb{N}$. If there exist $n_0, p \in \mathbb{N}$, $p > 0$, such that for all $x \geq n_0$, $x \in I \iff x + p \in I$, then we say that I is *ultimately periodic* (u.p.). It is known that if I is ultimately periodic, then $\{x \in \Sigma^* \;:\; |x| \in I\}$ is regular for any alphabet Σ.

Given alphabets Σ, Δ, a *morphism* is a function $h : \Sigma^* \to \Delta^*$ satisfying $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$. Given a morphism $h : \Sigma^* \to \Delta^*$ and a language $L \subseteq \Sigma^*$, then the image of L under h is given by $h(L) = \{h(x) \;:\; x \in L\}$, while if $L' \subseteq \Delta^*$, the inverse image of L' under h is defined by $h^{-1}(L') = \{x \in \Sigma^* \;:\; h(x) \in L'\}$.

We recall the definition of shuffle on trajectories, originally given by Mateescu *et al.* [19]. Shuffle on trajectories is defined by first defining the shuffle of two words x and y over an alphabet Σ on a trajectory t, which is simply a word in $\{0, 1\}^*$. We denote the shuffle of x and y along trajectory t by $x \sqcup\!\sqcup_t y$.

If $x = ax'$, $y = by'$ (with $a, b \in \Sigma$) and $t = et'$ (with $e \in \{0, 1\}$), then

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

If $x = ax'$ $(a \in \Sigma)$, $y = \epsilon$ and $t = et'$ $(e \in \{0, 1\})$, then

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

If $x = \epsilon$, $y = by'$ $(b \in \Sigma)$ and $t = et'$ $(e \in \{0, 1\})$, then

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

We let $x \sqcup\!\sqcup_\epsilon y = \emptyset$ if $\{x, y\} \neq \{\epsilon\}$. Finally, if $x = y = \epsilon$, then $\epsilon \sqcup\!\sqcup_t \epsilon = \epsilon$ if $t = \epsilon$ and $\emptyset$ otherwise.

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

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

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

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

Also, if $x = ax'$ $(a \in \Sigma)$ and $t = et'$ $(e \in \{i, d\})$, then

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

If $x \neq \epsilon$, then $x \leadsto_\epsilon y = \emptyset$. Further, $\epsilon \leadsto_t y = \epsilon$ if $t = y = \epsilon$. Otherwise, $\epsilon \leadsto_t y = \emptyset$.

Example 2.1 *Let $x = abcabc$, $y = bac$ and $t = (id)^3$. Then we have that $x \leadsto_t y = acb$. If $t = i^2 d^3 i$ then $x \leadsto_t y = \emptyset$.*

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

$$x \leadsto_T y = \bigcup_{t \in T} x \leadsto_t y.$$

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

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

Note that $\leadsto_T$ is neither an associative nor a commutative operation on languages, in general. We consider the following examples of deletion along trajectories:

(a) if $T = i^* d^*$, then $\leadsto_T = /$, the right-quotient operation;
(b) if $T = d^* i^*$, then $\leadsto_T = \backslash$, the left-quotient operation;
(c) if $T = i^* d^* i^*$, then $\leadsto_T = \rightarrow$, the deletion operation (see, e.g., Kari [11,12]);

(d) if $T = (i+d)^*$, then $\rightsquigarrow_T = \rightsquigarrow$, the scattered deletion operation (see, e.g., Ito *et al.* [10]);

(e) if $T = d^* i^* d^*$, then $\rightsquigarrow_T = \rightleftharpoons$, the bi-polar deletion operation (see, e.g., Kari [12]).

(f) let $k \geq 0$ and $T_k = i^* d^* i^{\leq k}$. Then $\rightsquigarrow_{T_k} = \rightarrow^k$, the k-deletion operation (see, e.g., Kari and Thierrin [13]).

Also, we note the difference between deletion along trajectories from the operation *splicing on routes* defined by Mateescu [18], which is a generalization of shuffle on trajectories which allows discarding symbols from either input word. Splicing on routes serves to generalize the *crossover operation* used in DNA computing by restricting the manner in which it may combine symbols, in a manner similar to how shuffle on trajectories restricts the way in which the shuffle operator may combine symbols (see Mateescu [18] for details and a definition of the crossover operation). Recently, we have shown that splicing on routes may be simulated by a fixed combination of shuffle and deletion along trajectories [5].

3 Closure and Characterization Results

The following lemma is proven by a direct construction:

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

PROOF. Let M_1, M_2, M_T be DFAs for L_1, L_2, T, respectively, with

$$M_j = (Q_j, \Sigma, \delta_j, q_j, F_j), \quad \text{for } j = 1, 2,$$
$$M_T = (Q_T, \{i, d\}, \delta_T, q_T, F_T).$$

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

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

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

$$\delta([q_j, q_k, q_\ell], \epsilon) = \{[\delta_1(q_j, a), \delta_2(q_k, a), \delta_T(q_\ell, d)] \ : \ a \in \Sigma\}$$

for all $[q_j, q_k, q_\ell] \in Q_1 \times Q_2 \times Q_T$. We can verify that M accepts $L_1 \rightsquigarrow_T L_2$. $\square$

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 (for the definitions of context-free languages (CFLs) and linear

CFLs, see, e.g., Hopcroft and Ullman [9]):

Theorem 3.2 *There exist languages L_1, L_2 and a set of trajectories $T \subseteq \{i,d\}^*$ satisfying each of the following:*

(a) L_1 is a CFL, L_2 is a singleton and T is regular, but $L_1 \rightsquigarrow_T L_2$ is 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:

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

Note that L_2 is a non-regular (linear) CFL. With these languages, we get that $L_1 \rightsquigarrow_T L_2 = L_2$. 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.

Recall that a *weak coding* is a morphism $\pi : \Sigma^* \to \Delta^*$ such that $\pi(a) \in \Delta \cup \{\epsilon\}$ for all $a \in \Sigma$. We have the following characterization of deletion along trajectories:

Theorem 3.3 *Let Σ be an alphabet. There exist weak codings $\rho_1, \rho_2, \tau, \varphi$ 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 $\hat{\Sigma} = \{\hat{a} \ : \ a \in \Sigma\}$ be a copy of Σ. Define the morphism $\rho_1 : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as follows: $\rho_1(\hat{a}) = \rho_1(a) = a$ for all $a \in \Sigma$ and $\rho_1(i) = \rho_1(d) = \epsilon$. Define $\rho_2 : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as follows: $\rho_2(\hat{a}) = a$ for all $a \in \Sigma$, $\rho_2(a) = \epsilon$ for all $a \in \Sigma$ and $\rho_2(d) = \rho_2(i) = \epsilon$.

Define $\tau : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as follows: $\tau(\hat{a}) = \tau(a) = \epsilon$ for all $a \in \Sigma$, $\tau(i) = i$ and $\tau(d) = d$. We define $\varphi : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \Sigma^*$ as $\varphi(\hat{a}) = \epsilon$ for all $a \in \Sigma$, $\varphi(a) = a$ for all $a \in \Sigma$, and $\varphi(i) = \varphi(d) = \epsilon$. Finally, we note that the result can be proven by letting $R = (i\Sigma + d\hat{\Sigma})^*$. $\square$

Recall that a *cone* (or *full trio*) is a class of languages closed under morphism, inverse morphism and intersection with regular languages [20, Sect. 3]. Thus, we have the following corollary:

Corollary 3.4 *Let $\mathcal{L}$ be a cone. 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 [9, Thm. 11.3] is a specific instance of Corollary 3.4. Lemma 3.1 can also be proven by appealing to Theorem 3.3. 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.*

The following result shows that if any of the conditions of Corollary 3.5 are not met, the result might not hold:

Theorem 3.6 *There exist languages L_1, L_2 and a set of trajectories $T \subseteq \{i, d\}^*$ satisfying each of the following:*

(a) L_1, L_2 are (linear) CFLs and T is regular, but $L_1 \rightsquigarrow_T L_2$ in not a CFL;
(b) L_1, T are (linear) CFLs, and L_2 is a singleton, but $L_1 \rightsquigarrow_T L_2$ is not a CFL;
(c) L_1 is regular, L_2, T are (linear) CFLs, but $L_1 \rightsquigarrow_T L_2$ is not a CFL.

PROOF. (a) The result is immediate, since it is known (see, e.g., Ginsburg and Spanier [7, Thm. 3.4]) that the CFLs are not closed under right quotient (given by the trajectory $T = i^*d^*$). The languages described by Ginsburg and Spanier which witness this non-closure are linear CFLs.

(b) 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 linear 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.

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

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

Then we can verify that $L_1 \rightsquigarrow_T L_2 = \{a^n b^n c^n : n \geq 0\}$, which is not a CFL. This completes the proof. □

Note that the context-sensitive languages (CSLs, see, e.g., Mateescu and Salomaa [20, Sect. 2]) are not a cone, since they are not closed under arbitrary morphism. Thus, Corollary 3.4 does not apply to the CSLs. In fact, it is known (see, e.g., Mateescu and Salomaa [20, Thm. 2.12]) that the CSLs are not closed under quotient with regular languages.

3.1 Recognizing Deletion Along Trajectories

We now consider the problem of giving a monoid recognizing deletion along trajectories, when the languages and set of trajectories under consideration are regular. Harju *et al.* [8] give a monoid which recognizes $L_1 \sqcup\!\sqcup_T L_2$ when L_1, L_2 and T are regular.

For a background on recognition of formal languages by monoids, consult Pin [24]. Let $L \subseteq \Sigma^*$ be a language. We say that a monoid M recognizes L if there exist 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 [24, p. 17]):

Theorem 3.7 *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 $\varphi_j : \Sigma^* \to M_i$ for $j = 1, 2$, $\varphi_T : \{i,d\}^* \to M_T$ and subsets F_1, F_2, F_T, respectively.

As in Harju *et al.* [8], 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 $y \in \Sigma^*$,

$$\varphi(y)D = \{[\varphi_1(\alpha), \varphi_2(\beta), \varphi_T(t)] \ : \ y \in \alpha \rightsquigarrow_t \beta, \alpha, \beta \in \Sigma^*, t \in \{i,d\}^*\}. \quad (1)$$

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 (1), we have that

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

This establishes the following result:

Lemma 3.8 *Let L_j be a regular language recognized by M_j for $j = 1, 2$ and $T \subseteq \{i,d\}^*$ be a regular set of trajectories recognized by the monoid M_T. Then $\mathcal{P}(M_1 \times M_2 \times M_T)$ recognizes $L_1 \rightsquigarrow_T L_2$.*

Thus, Lemma 3.8 gives another proof of Lemma 3.1.

3.2 *Equivalence of Trajectories*

We briefly note that two sets of trajectories define the same operation if and only if they are equal. More precisely, if $T_1, T_2 \subseteq \{i, d\}^*$, say that T_1 and T_2 are *equivalent* if $L_1 \rightsquigarrow_{T_1} L_2 = L_1 \rightsquigarrow_{T_2} L_2$ for all languages L_1, L_2.

Lemma 3.9 *Let $T_1, T_2 \subseteq \{i, d\}^*$. Then T_1, T_2 are equivalent iff $T_1 = T_2$.*

PROOF. If $T_1 = T_2$ then clearly T_1 and T_2 are equivalent. If T_1 and T_2 are not equal, then without loss of generality, let $t \in T_1 - T_2$. Let $n = |t|_i$ and $m = |t|_d$. Then it is not hard to see that $i^n \in \{t\} \rightsquigarrow_{T_1} \{d^m\}$, but that $i^n \notin \{t\} \rightsquigarrow_{T_2} \{d^m\}$, i.e., T_1 and T_2 are not equivalent. $\square$

Thus, for instance, it is decidable whether T_1, T_2 are equivalent if, e.g., T_1 is regular and T_2 is an unambiguous CFL, but undecidable if T_1 is regular and T_2 is an arbitrary CFL.

4 Regularity-Preserving Non-Regular Trajectories

Consider the following result of Mateescu *et al.* [19, 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 sets of trajectories $\mathcal{C}$, which contains non-regular languages, such that for all regular languages R_1, R_2, and for all $H \in \mathcal{C}$, $R_1 \rightsquigarrow_H R_2$ 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 $H = \{i^n d^n \ : \ n \geq 0\}$. Note that

$$R_1 \rightsquigarrow_H R_2 = \{x \in \Sigma^* \ : \ \exists y \in R_2 \text{ such that } xy \in R_1 \text{ and } |x| = |y|\}.$$

We can establish directly (by constructing an NFA) that for all regular languages $R_1, R_2 \subseteq \Sigma^*$, the language $R_1 \rightsquigarrow_H R_2$ is regular. However, H is a non-regular CFL.

Remark that $R_1 \rightsquigarrow_H R_2$ is similar to proportional removals studied by Stearns and Hartmanis [26], Amar and Putzolu [1,2], Seiferas and McNaughton [25], Kosaraju [15,16], Kozen [17], Zhang [28], the author [4] and others. In particular, we note the case of $\frac{1}{2}(L)$, given by

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

Thus, $\frac{1}{2}(L) = L \leadsto_H \Sigma^*$. 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.

Call a relation $r \subseteq \mathbb{N}^2$ *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 binary relations r that preserve regularity are precisely the u.p.-preserving relations [25].

We note the inclusion

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

holds for $H = \{i^n d^n \ : \ n \geq 0\}$. However, equality does not hold in general. Consider the languages $L_1 = \{0^2, 0^4\}$, $L_2 = \{0^3\}$. Then $0 \in \frac{1}{2}(L_1) \cap L_1/L_2$. However, $0 \notin L_1 \leadsto_H L_2$. Thus, we note that

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

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. We have the following result:

Theorem 4.1 *Let $r \subseteq \mathbb{N}^2$ be a binary relation and $H_r = \{i^n d^m \ : \ r(n, m)\}$. The operation $\leadsto_{H_r}$ is regularity-preserving iff r is u.p.-preserving.*

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

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

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

For the reverse implication, we modify the construction of Seiferas and McNaughton [25, Thm. 1]. Let L_1, L_2 be regular, and let $M_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$ be the minimal complete DFA for L_1. Then, for each $q \in Q_1$, we let $L_1^{(q)}$ be the language accepted by the DFA $M_1^{(q)} = (Q_1, \Sigma, \delta_1, q_0, \{q\})$. Let R_q be the language accepted by the DFA $N_1^{(q)} = (Q_1, \Sigma, \delta_1, q, F_1)$. Note that $L_1^{(q)} = \{w \in \Sigma^* : \delta(q_0, w) = q\}$ and $R_q = \{w \in \Sigma^* : \delta(q, w) \in F_1\}$.

As M_1 is complete, $\Sigma^* = \bigcup_{q \in Q_1} L_1^{(q)}$. Thus,

$$L_1 \rightsquigarrow_{H_r} L_2 = \bigcup_{q \in Q_1} (L_1 \rightsquigarrow_{H_r} L_2) \cap L_1^{(q)}.$$

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

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

It is easy to see 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. □

Note that, in general,

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

Consider the following 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 [28, Sect. 3]). However, H_e is not CF. The set H_e is, however, a linear conjunctive language (see Okhotin [23] 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 [28, Thm. 5.1]). However, H_f is not a CFL, nor a linear conjunctive language [23].

Thus, there are non-CF trajectories which preserve regularity. Kozen states that there are even H_r which preserve regularity but are "highly noncomputable" [17, p. 3].

We can extend the class of non-regular sets of trajectories T such that $L_1 \leadsto_T L_2$ is regular for all regular languages L_1, L_2 by considering T such that $T \subseteq (d^* i^*)^m d^*$ for some $m \geq 1$ [2]. To consider such non-regular T, it will be advantageous to adopt the notations of Zhang [28] on *boolean matrices*. We summarize these notions below; for a full review, the reader may consult the original paper.

For any finite set Q, let $\mathcal{M}(Q)$ denote the set of square Boolean matrices indexed by Q. Let $\mathcal{V}(Q)$ denote the set of Boolean vectors indexed by Q. For an automaton over a set of states Q, we will associate with it matrices from $\mathcal{M}(Q)$ and vectors from $\mathcal{V}(Q)$.

In particular, let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA. Then for each $a \in \Sigma$, let $\nabla_a \in \mathcal{M}(Q)$ be the matrix defined by transitions on a, that is $\nabla_a(q_1, q_2) = 1$ iff $\delta(q_1, a) = q_2$. Let $\nabla = \sum_{a \in \Sigma} \nabla_a$ (where addition is taken to be Boolean addition, i.e., $0 + 0 = 0$, $0 + 1 = 1 + 0 = 1 + 1 = 1$). Thus, note that $\nabla(q_1, q_2) = 1$ iff there is some $a \in \Sigma$ such that $\delta(q_1, a) = q_2$. Note that taking powers of ∇ yields information on paths of different lengths: for all $i \geq 0$, $\nabla^i(q_1, q_2) = 1$ iff there is a path of length i from q_1 to q_2.

For any $Q' \subseteq Q$, let $I_{Q'} \in \mathcal{V}(Q)$ be the characteristic vector of Q', given by $I_{Q'}(q) = 1$ iff $q \in Q'$. If Q' is a singleton q, we denote $I_{\{q\}}$ by I_q. Note that if $Q_1, Q_2 \subseteq Q$ and $i \geq 0$, then $I_{Q_1} \cdot \nabla^i \cdot I_{Q_2}^t = 1$ iff there is a path of length i from some state in Q_1 to some state in Q_2 (here, I^t denotes the transpose of I).

Call a function $f : \mathbb{N} \to \mathbb{N}$ *ultimately periodic with respect to powers of Boolean matrices* [28], abbreviated m.u.p. (for "matrix ultimately periodic"), if, for all square Boolean matrices ∇, there exist natural numbers e, p ($p > 0$) such that for all $n \geq e$,

$$\nabla^{f(n)} = \nabla^{f(n+p)}.$$

Let $m \geq 1$. We will define a class of $T \subseteq (d^* i^*)^m d^*$ such that for all regular languages R_1, R_2, $R_1 \leadsto_T R_2$ is regular. In particular, let $m \geq 1$, and let $f_\ell^{(j)} : \mathbb{N} \to \mathbb{N}$ be a m.u.p. function for each $1 \leq \ell \leq m+1$ and $1 \leq j \leq m$.

[2] The choice of $T \subseteq (d^* i^*)^m d^*$ rather than, e.g., $T \subseteq (i^* d^*)^m$ or $T \subseteq (d^* i^*)^m$ is arbitrary. The same type of formulation and arguments can be applied to these similar types of sets of trajectories.

Define $X_\ell : \mathbb{N}^m \to \mathbb{N}$ for $1 \le \ell \le m+1$ by

$$X_\ell(n_1, n_2, \ldots, n_m) = \sum_{j=1}^{m} f_\ell^{(j)}(n_j).$$

We will use the abbreviation $\vec{n} = (n_1, n_2, \ldots, n_m)$. Finally, we define

$$T = \{\prod_{j=1}^{m} (d^{X_j(\vec{n})} i^{n_j}) d^{X_{m+1}(\vec{n})} \ : \ \vec{n} = (n_1, \ldots, n_m) \in \mathbb{N}^m\}. \qquad (2)$$

The set T satisfies our intuition that the 'i-portions' may not interact with each other, but may interact with any 'd-portion' they wish to. Our claim that these T preserve regularity is proven in the following theorem.

Theorem 4.2 *Let $m \ge 1$, and $f_\ell^{(j)}$ be m.u.p. for $1 \le \ell \le m+1$ and $1 \le j \le m$. Let $T \subseteq (d^* i^*)^m d^*$ be defined by (2). Then for all regular languages R_1, R_2, the language $R_1 \rightsquigarrow_T R_2$ is regular.*

Let $\mathbf{m} = \{0, 1, 2, 3, \ldots, m\}$ for any $m \ge 1$.

PROOF. Let $M_i = (Q_i, \Sigma, \delta_i, s_i, F_i)$ be a DFA accepting R_i for $i = 1, 2$. Let $M_{1,2} = (Q_1 \times Q_2, \Sigma, \delta_0, (s_1, s_2), F_1 \times F_2)$ where δ is given by $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$ for all $(q_1, q_2) \in Q_1 \times Q_2$ and all $a \in \Sigma$. Note that $M_{1,2}$ accepts $R_1 \cap R_2$. Let ∇ be the adjacency matrix for $M_{1,2}$. For each $1 \le j \le m$ and $1 \le \ell \le m+1$, let $e_\ell^{(j)} \ge 0$ and $p_\ell^{(j)} > 0$ be chosen so that $\nabla^{f_\ell^{(j)}(n)} = \nabla^{f_\ell^{(j)}(n + p_\ell^{(j)})}$ for all $n \ge e_\ell^{(j)}$.

For all $1 \le j \le m$ and $1 \le \ell \le m+1$, let $g_\ell^{(j)} = e_\ell^{(j)} + p_\ell^{(j)}$, and define the set

$$\mathfrak{M}(j, \ell) = \{\nabla^{f_\ell^{(j)}(i)} \ : \ 0 \le i \le e_\ell^{(j)} + p_\ell^{(j)}\} \times \mathbf{g}_\ell^{(\mathbf{j})}.$$

We will define an NFA $M = (Q, \Sigma, \delta, S, F)$ which we claim accepts $R_1 \rightsquigarrow_T R_2$. The NFA will be nondeterministic, and will also have multiple start states. Our state set Q is given by

$$Q = \mathbf{m} \times (\prod_{\ell=1}^{m} (\prod_{j=1}^{m} \mathfrak{M}(j, \ell)) \times Q_1^3 \times Q_2) \times \prod_{j=1}^{m} \mathfrak{M}(j, m+1).$$

Let $\mu_{j,\ell} = [\nabla^{f_\ell^{(j)}(0)}, 0] \in \mathfrak{M}(j, \ell)$. Our set S of initial states is given by

$$S = \{1\} \times (\prod_{\ell=1}^{m} \prod_{j=1}^{m} \mu_{j,\ell} \times \{[q, q] \ : \ q \in Q_1\} \times Q_1 \times Q_2) \times \prod_{j=1}^{m} \mu_{j,m+1}.$$

To partially motivate this definition, the elements of the form Q_1^3 will represent one path through M_1: the first element will represent our nondeterministic "guess" of where the path starts, the second state will actually trace the path through M_1 (along a portion of our input word) and the third state represents our guess of where the path will end. Thus, during the course of our computation, the first and third elements are never changed; only the second is affected by the input word. The first and third elements are used to verify (once the computation has completed) that our guesses for the start and finish are correct, and that they correspond ("match up") with the guessed paths for the adjacent components. The elements of Q_2 will represent our guesses of the intermediate points of the path through M_2; similarly to our guesses in Q_1, it will not change through the computation.

Our set of final states F is given by those states of the form

$$\{m\} \times \left(((A_\ell^{(j)}, c_\ell^{(j)})_{j=1}^m q_\ell^{(1)}, q_\ell^{(2)}, q_\ell^{(3)}, r_\ell)_{\ell=1}^m, (A_{m+1}^{(j)}, c_\ell^{(j)})_{j=1}^m\right),$$

where the following conditions are met:

(F-i) for all $1 \leq \ell \leq m$, $I_{(q_{\ell-1}^{(3)}, r_{\ell-1})} \cdot (\prod_{j=1}^m A_\ell^{(j)}) \cdot I^t_{(q_\ell^{(1)}, r_\ell)} = 1$ (we let $q_0^{(3)} = s_1$, the start state of M_1 and $r_0 = s_2$ the start state of M_2);

(F-ii) $I_{(q_m^{(3)}, r_m)} \cdot (\prod_{j=1}^m A_{m+1}^{(j)}) \cdot I^t_{F_1 \times F_2} = 1$;

(F-iii) for all $1 \leq \ell \leq m$, we have $q_\ell^{(2)} = q_\ell^{(3)}$.

We will see that the matrix $A_\ell^{(j)}$ will ensure that there is a path of length $f_\ell^{(j)}(n_j)$ through $M_1 \times M_2$. Thus, condition (F-i) will ensure that we have a path from our guessed end state of the previous i-portion through to the guessed start state of the next i-portion. This will correspond to the presence of some word w of length $\sum_{j=1}^m f_\ell^{(j)}(n_j)$ which takes M from the end state of the previous i-portion to the start of the next i-portion. The condition (F-ii) will ensure that the final d-portion ends in a final state in both M_1 and M_2.

Condition (F-iii) verifies that the nondeterministic "guesses" for the end of each i-portion path is correct.

Finally, we may define the action of δ. We will adopt the convention of Zhang [28] and denote by $\langle c \rangle_a^b$ the quantity

$$\langle c \rangle_a^b = \begin{cases} c & \text{if } c \leq a; \\ a + ((c-a) \bmod b\ ) & \text{otherwise.} \end{cases}$$

Further, to describe the action of δ more easily, we introduce auxiliary functions $\Upsilon_{\ell,\alpha}$ for all $1 \leq \ell \leq m+1$ and $1 \leq \alpha \leq m$. In particular

$$\Upsilon_{\ell,\alpha} : \prod_{j=1}^{m} \mathfrak{M}(j,\ell) \to \prod_{j=1}^{m} \mathfrak{M}(j,\ell)$$

is given by

$$\begin{aligned} &\Upsilon_{\ell,\alpha}((\nabla^{f_\ell^{(j)}(c_\ell^{(j)})}, c_\ell^{(j)})_{j=1}^{m}) \\ =&((\nabla^{f_\ell^{(j)}(c_\ell^{(j)})}, c_\ell^{(j)})_{j=1}^{\alpha-1}, \nabla^{f_\ell^{(\alpha)}(\langle c_\ell^{(\alpha)}+1\rangle^{p_\ell^{(\alpha)}}_{e_\ell^{(\alpha)}})}, \langle c_\ell^{(\alpha)}+1\rangle^{p_\ell^{(\alpha)}}_{e_\ell^{(\alpha)}}, (\nabla^{f_\ell^{(j)}(c_\ell^{(j)})}, c_\ell^{(j)})_{j=\alpha+1}^{m}). \end{aligned}$$

Note that $\Upsilon_{\ell,\alpha}$ updates the α-th component, while leaving all other components unchanged.

Then we define δ by

$$\begin{aligned} &\delta\left(\left(\alpha, ((\nabla^{f_\ell^{(j)}(c_\ell^{(j)})}, c_\ell^{(j)})_{j=1}^{m}, p_\ell^{(1)}, p_\ell^{(2)}, p_\ell^{(3)}, r_\ell)_{\ell=1}^{m}, (\nabla^{f_{m+1}^{(j)}(c_{m+1}^{(j)})}, c_{m+1}^{(j)})_{j=1}^{m}\right), a\right) \\ &= \Big\{ \Big(\alpha+\beta, (\Upsilon_{\ell,\alpha+\beta}((\nabla^{f_\ell^{(j)}(c_\ell^{(j)})}, c_\ell^{(j)})_{j=1}^{m}), p_\ell^{(1)}, p_\ell^{(2)}, p_\ell^{(3)}, r_\ell)_{\ell=1}^{\alpha+\beta-1}, \\ &\quad \Upsilon_{\alpha+\beta,\alpha+\beta}((\nabla^{f_{\alpha+\beta}^{(j)}(c_{\alpha+\beta}^{(j)})}, c_{\alpha+\beta}^{(j)})_{j=1}^{m}), p_{\alpha+\beta}^{(1)}, \delta_1(p_{\alpha+\beta}^{(2)}, a), p_{\alpha+\beta}^{(3)}, r_{\alpha+\beta} \\ &\quad (\Upsilon_{\ell,\alpha+\beta}((\nabla^{f_\ell^{(j)}(c_\ell^{(j)})}, c_\ell^{(j)})_{j=1}^{m}), p_\ell^{(1)}, p_\ell^{(2)}, p_\ell^{(3)}, r_\ell)_{\ell=\alpha+\beta+1}^{m}, \\ &\quad \Upsilon_{m+1,\alpha+\beta}((\nabla^{f_{m+1}^{(j)}(c_{m+1}^{(j)})}, c_{m+1}^{(j)})_{j=1}^{m})\Big) : 0 \leq \beta \leq m-\alpha \Big\}. \end{aligned}$$

Note that, though the definition of δ is complicated, its action is straightforward. The index α indicates the 'i-portion' which is currently receiving the input. Given that we are currently in the α-th i-portion, we may nondeterministically choose to move to any of the subsequent portions. The action of the function $\Upsilon_{\ell,\alpha}$ is to simulate the corresponding function f_ℓ^{α}.

We show that $L(M) \subseteq R_1 \rightsquigarrow_T R_2$. If we arrive at a final state, by (F-i), for each $1 \leq \ell \leq m$ there is a word x_ℓ of length $X_\ell(\vec{n})$ which takes us from state $q_{\ell-1}^{(3)}$ to $q_\ell^{(1)}$ in M_1 and also takes us from $r_{\ell-1}$ to r_ℓ in M_2. By the choice of S, δ and condition (F-iii), for each $1 \leq \ell \leq m$, there is a word w_i of length n_i which takes us from state $q_\ell^{(1)}$ to $q_\ell^{(3)}$. Further, the input word is of the form $w = w_1 w_2 \cdots w_m$. Finally, by (F-ii), there is a word x_{m+1} of length $X_{m+1}(\vec{n})$ which takes us from state q_m to a final state in M_1 and from r_m to a final state in M_2. The situation is illustrated in Figure 1.

Thus, we conclude that $x_1 w_1 \cdots x_m w_m x_{m+1} \in R_1$, $x_1 \cdots x_{m+1} \in R_2$ and $|x_\ell| = X_\ell(\vec{n})$ for all $1 \leq \ell \leq m+1$. Thus, $w_1 \cdots w_m \in R_1 \rightsquigarrow_T R_2$. A similar argument, which is left to the reader, shows the reverse inclusion. $\Box$

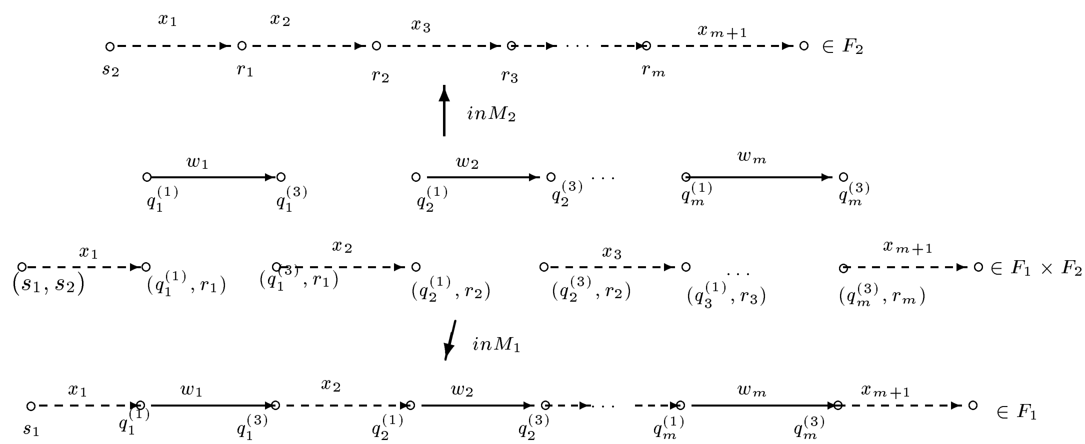

Fig. 1. Construction of the words in M_1 and M_2 from the action of M.

As an example, consider $m = 1$ and let $f_1^{(1)}, f_2^{(2)}$ both be the identity function. Then the conditions of Theorem 4.2 are met and $T = \{d^n i^n d^n \ : \ n \geq 0\}$. Consider then that

$$R_1 \rightsquigarrow_T \Sigma^* = \{x \ : \ \exists y, z \in \Sigma^{|x|} \text{ such that } yxz \in R_1\}.$$

This is the 'middle-thirds' operation, which is sometimes used as a challenge problem for undergraduates in formal language theory (see, e.g., Hopcroft and Ullman [9, Ex. 3.17]). We may immediately conclude that the regular languages are preserved under the middle-thirds operation.

We note that the condition that $(n_1, n_2, \ldots, n_m) \in \mathbb{N}^m$ in (2) can be replaced by the conditions that, for all $1 \leq j \leq m$, $n_j \in I_j$ for an arbitrary u.p. set $I_j \subseteq \mathbb{N}$. The construction adds considerable detail to the proof of Theorem 4.2, and is omitted. With this extension, we can also consider a class of examples given by Amar and Putzolu [2], which are equivalent to trajectories of the form

$$AP(k_1, k_2, \alpha) = \{i^{mk_1} d^{mk_2 + \alpha} \ : \ m \geq 0\},$$

for fixed $k_1, k_2, \alpha \geq 0$ with $\alpha < k_1 + k_2$. For any $k_1, k_2, \alpha \geq 0$, we can conclude that the operation $\rightsquigarrow_A$ preserves regularity, where $A = AP(k_1, k_2, \alpha)$. This was established by Amar and Putzolu [2] by means of *even linear grammars*.

5 Deletion as an Inverse of Shuffle on Trajectories

In this section, we show that deletion along trajectories constitutes the inverse of shuffle on trajectories, in the sense introduced by Kari [12]. We then show how this implies several positive decidability results regarding equations involving shuffle on trajectories (undecidability results will be examined in a forthcoming paper).

Given two binary word operations $\diamond, \star : (\Sigma^*)^2 \to 2^{\Sigma^*}$, we say that $\diamond$ is a left-inverse of $\star$ [12, Defn. 4.1] if, for all $u, v, w \in \Sigma^*$,

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

Let $\tau : \{0,1\}^* \to \{i,d\}^*$ be the morphism given by $\tau(0) = i$ and $\tau(1) = d$. Then we have the following characterization of left-inverses:

Theorem 5.1 *Let $T \subseteq \{0,1\}^*$ be a set of trajectories. Then $ш_T$ and $\rightsquigarrow_{\tau(T)}$ are left-inverses of each other.*

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

$$\epsilon \in u ш_t v \iff u = v = t = \epsilon \iff u \in (\epsilon \rightsquigarrow_{\tau(t)} v).$$

Let $w \in \Sigma^+$ and assume that the result is true for all words shorter than w. Let $w = aw'$ for $a \in \Sigma$.

First, assume that $aw' \in u ш_t v$. As $|t| = |w|$, we have that $t \neq \epsilon$. Let $t = et'$ for some $e \in \{0,1\}$. There are two cases:

(a) If $e = 0$, then we have that $u = au'$ and that $w' \in u' ш_{t'} v$. By induction, $u' \in w' \rightsquigarrow_{\tau(t')} v$. Thus,

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

(b) If $e = 1$, then we have that $v = av'$ and $w' \in u ш_{t'} v'$. By induction, $u \in w' \rightsquigarrow_{\tau(t')} v'$. Thus,

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

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

Now, let us assume that $u \in w \rightsquigarrow_{\tau(t)} v$. As $|t| = |\tau(t)| = |w| \geq 1$, let $t = et'$ for some $e \in \{0,1\}$. We again have two cases:

(a) If $e = 0$, then $\tau(e) = i$. Then necessarily $u = au'$, and $u' \in w' \rightsquigarrow_{\tau(t')} v$. By induction $w' \in u' ш_{t'} v$. Thus,

$$\begin{aligned}(u \sqcup\!\sqcup_t v) &= (au' \sqcup\!\sqcup_{0t'} v) \\ &= a(u' \sqcup\!\sqcup_{t'} v) \ni aw' = w.\end{aligned}$$

(b) If $e = 1$, then $\tau(e) = d$. Then necessarily $v = av'$, and $u \in (w' \rightsquigarrow_{\tau(t')} v')$. By induction, $w' \in u \sqcup\!\sqcup_{t'} v'$. Thus,

$$\begin{aligned}(u \sqcup\!\sqcup_t v) &= (u \sqcup\!\sqcup_{1t'} av') \\ &= a(u \sqcup\!\sqcup_{t'} v') \ni aw' = w.\end{aligned}$$

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

We note that Theorem 5.1 agrees with the observations of Kari [12, 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 [12, 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 [12, 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 and the constructions are effective, 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 $\diamond, \star : (\Sigma^*)^2 \to 2^{\Sigma^*}$, we say that $\diamond$ is a right-inverse [12, Defn. 4.1] 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* $\diamond$ [12].

Let $\pi : \{0,1\}^* \to \{i,d\}^*$ be the morphism given by $\pi(0) = d$ and $\pi(1) = i$. We can repeat the above arguments for right-inverses instead of left-inverses:

Theorem 5.5 *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.*

PROOF. Let $sym_s : \{0,1\}^* \to \{0,1\}^*$ be the morphism given by $sym_s(0) = 1$ and $sym_s(1) = 0$. Then it is easy to note that

$$x \in u \sqcup\!\sqcup_t v \iff x \in v \sqcup\!\sqcup_{sym_s(t)} u.$$

Thus, using Theorem 5.1, we note that

$$\begin{aligned} x \in u \sqcup\!\sqcup_t v &\iff x \in v \sqcup\!\sqcup_{sym_s(t)} u \\ &\iff v \in x \rightsquigarrow_{\tau(sym_s(t))} u \\ &\iff v \in u (\rightsquigarrow_{\tau(sym_s(t))})^r x. \end{aligned}$$

Thus, the result follows on noting that $\pi \equiv \tau \circ sym_s$. □

This again agrees with the observations of Kari [12, 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 [3]. We will investigate the decidability of $R = X_1 \sqcup\!\sqcup_T X_2$, where T is a fixed set of trajectories, X_1, X_2 are unknown and R is regular, in a forthcoming paper.

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

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

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

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 [12].

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 $\{x\} \sqcup\!\sqcup_T L = R$.*

PROOF. Let $r = \min\{|y| \;:\; y \in R\}$. Given a DFA for R, it is clear that we can compute r by breadth-first search. Then note that $|z| = |x| + |y|$ for all $z \in x \sqcup\!\sqcup_T y$ (regardless of T). Thus, it is clear that if there exists $x \in \Sigma^*$ satisfying $\{x\} \sqcup\!\sqcup_T L = R$, then $|x| \leq r$. Our algorithm then simply considers all words x of length at most r, and checks whether $\{x\} \sqcup\!\sqcup_T L = R$ holds. $\square$

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 $sym_d : \{i,d\}^* \to \{i,d\}^*$ be the morphism given by $sym_d(i) = d$ and $sym_d(d) = i$.

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

PROOF. By Theorems 5.5 and 5.1, we note that

$$\begin{aligned} x \in y \rightsquigarrow_t z &\iff y \in x \sqcup\!\sqcup_{\tau^{-1}(t)} z \\ &\iff z \in y \rightsquigarrow_{\pi(\tau^{-1}(t))} x \end{aligned}$$

The result follows on noting that $\pi \circ \tau^{-1} \equiv sym_d$. $\square$

We note that Theorem 5.8 agrees with the observations of Kari [12, Obs. 4.4]. Also, we have 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 word in Σ^*, and L, R, T are regular languages. Equations of this form have previously been considered by Kari [12]. Our constructions generalize those of Kari directly.

We begin with the following technical lemma:

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

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

PROOF. Let x be a word 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 exist $y \in \overline{R}$, $z \in L$ 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 $y \in (\{x\} \rightsquigarrow_T L) \cap \overline{R}$. 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 words $z \in L$ and $t \in T$ such that

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

Let y be some word 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. $\square$

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 the set of trajectories T from a finite word x. Thus, assume that R is finite. Then we can construct the following regular language:

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

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

$$x \in \{x \ : \ \{x\} \rightsquigarrow_T L \subseteq R\}; \tag{3}$$
$$x \notin \{x \ : \ \{x\} \rightsquigarrow_T L \subseteq S \subsetneq R\}. \tag{4}$$

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

Thus, if R is finite, to decide if a word 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). $\square$

6 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-like operations which have been studied in the literature. The closure properties of deletion along trajectories 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 shown that a large class of sets of trajectories, which includes several operations known in the literature, define deletion operations which preserve regularity.

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

Acknowledgments

I would like to thank Kai Salomaa for reading several versions of this paper and offering many helpful suggestions and corrections.

I am grateful to the referees who made several helpful suggestions and pointed me to the work of Amar and Putzolu.

References

[1] V. Amar, G. Putzolu, On a family of linear grammars, Information and Control 7 (1964) 283–291.

[2] V. Amar, G. Putzolu, Generalizations of regular events, Information and Control 8 (1965) 56–63.

[3] C. Câmpeanu, K. Salomaa, S. Vágvölgyi, Shuffle decompositions of regular languages, International Journal of Foundations of Computer Science 13 (6) (2002) 799–816.

[4] M. Domaratzki, State complexity of proportional removals, Journal of Automata, Languages and Combinatorics 7 (4) (2002) 455–468.

[5] M. Domaratzki, Splicing on routes versus shuffle and deletion along trajectories, Tech. Rep. 2003–471, School of Computing, Queen's University at Kingston (2003).

[6] M. Domaratzki, K. Salomaa, State complexity of shuffle on trajectories, in: J. Dassow, M. Hoeberechts, H. Jürgensen, D. Wotschke (Eds.), Descriptional Complexity of Formal Systems (DCFS), 2002, pp. 81–94.

[7] S. Ginsburg, E. Spanier, Quotients of context-free languages, Journal of the ACM 10 (4) (1963) 487–492.

[8] T. Harju, A. Mateescu, A. Salomaa, Shuffle on trajectories: The Schützenberger product and related operations, in: MFCS 1998, no. 1450 in Lecture Notes in Computer Science, Springer-Verlag, 1998, pp. 503–511.

[9] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.

[10] M. Ito, L. Kari, G. Thierrin, Shuffle and scattered deletion closure of languages, Theoretical Computer Science 245 (2000) 115–133.

[11] L. Kari, Generalized derivatives, Fundamenta Informaticae 18 (1993) 27–39.

[12] L. Kari, On language equations with invertible operations, Theoretical Computer Science 132 (1994) 129–150.

[13] L. Kari, G. Thierrin, k-catenation and applications: k-prefix codes, Journal of Information and Optimization Sciences 16 (2) (1995) 263–276.

[14] L. Kari, G. Thierrin, Maximal and minimal solutions to language equations, Journal of Computer and System Sciences 53 (1996) 487–496.

[15] S. Kosaraju, Regularity preserving functions, ACM SIGACT News 6 (2) (1974) 16–17.

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

[17] D. Kozen, On regularity-preserving functions, Tech. Rep. TR95-1559, Department of Computer Science, Cornell University (1995).

[18] A. Mateescu, Splicing on routes: a framework of DNA computation, in: C. Calude, J. Casti, M. Dinneen (Eds.), Unconventional Models of Computation, Springer, 1998, pp. 273–285.

[19] A. Mateescu, G. Rozenberg, A. Salomaa, Shuffle on trajectories: Syntactic constraints, Theoretical Computer Science 197 (1998) 1–56.

[20] A. Mateescu, A. Salomaa, Aspects of classical language theory, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. I, Springer-Verlag, 1997, pp. 175–246.

[21] A. Mateescu, A. Salomaa, Nondeterministic trajectories, in: Formal and Natural Computing, Vol. 2300 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 96–106.

[22] A. Mateescu, K. Salomaa, S. Yu, On fairness of many-dimensional trajectories, Journal of Automata, Languages and Combinatorics 5 (2000) 145–157.

[23] A. Okhotin, On the equivalence of linear conjunctive grammars to trellis automata, RAIRO Theoretical Informatics and Applications 38 (2004) 69–88.

[24] J.-E. Pin, Varieties of Formal Languages, Plenum, 1986.

[25] J. Seiferas, R. McNaughton, Regularity-preserving relations, Theoretical Computer Science 2 (1976) 147–154.

[26] R. Stearns, J. Hartmanis, Regularity preserving modifications of regular expressions, Information and Control 6 (1) (1963) 55–69.

[27] S. Yu, Regular languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. I, Springer-Verlag, 1997, pp. 41–110.

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