Chapter 5

Deletion along Trajectories

5.1 Introduction

As we have seen, shuffle on trajectories is a powerful method for unifying operations which insert all the letters of one word into another. Concurrent to this research, Kari and others [106, 117] have done research into the inverses of insertion- and shuffle-like operations, which have yielded decidability results for language 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 *bi-polar deletion* [106].

In this chapter, we introduce the notion of *deletion along trajectories*, which is the analogous notion to 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 5.8), defined by a deletion along a renaming of T. This yields the result that it is decidable whether language equations of the form $L ⧢_T X = R$ 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 5.4.1 and 5.4.2 explicitly define classes of sets of trajectories, which include non-regular sets, which preserve regularity.

5.2 Definitions

We now give the main definition of this chapter, 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 $x \rightsquigarrow_t y$, the deletion of y from x along trajectory t, 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, if $x = ax'$ ($a \in \Sigma$) and $t = et'$ ($e \in \{i, d\}$), then

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

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

Example 5.2.1: Let $x = abcabc$, $y = bac$ and $t = (id)^3$. Then we have that $x \rightsquigarrow_t y = acb$. If $t = i^2d^3i$ then $x \rightsquigarrow_t y = \emptyset$. □

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

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

Note that $\rightsquigarrow_T$ is neither an associative nor a commutative operation on languages, in general. We consider the following examples of deletion along trajectories (for any operations not defined, we refer the reader to the appropriate paper cited below):

(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 $\rightsquigarrow_T = \rightarrow$, the deletion operation (see, e.g., Kari [103, 106]);

(d) if $T = (i + d)^*$, then $\rightsquigarrow_T = \rightsquigarrow$, the scattered deletion operation (see, e.g., Ito *et al.* [79]);

(e) if $T = d^*i^*d^*$, then $\rightsquigarrow_T = \rightleftharpoons$, the bi-polar deletion operation (see, e.g., Kari [106]);

(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 [114]).

Also, we note the difference between deletion along trajectories from the operation *splicing on routes* defined by Mateescu [144], which is a generalization of shuffle on trajectories which allows discarding letters 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 letters, in a manner similar to how shuffle on trajectories restricts the way in which the shuffle operator may combine letters (see Mateescu [144] for details and a definition of the crossover operation).

5.3 Closure and Characterization Results

The following lemma is proven by a direct construction:

Lemma 5.3.1 *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

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

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 $L_1 \rightsquigarrow_T L_2$. ■

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

Theorem 5.3.2 *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 an LCFL.

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 have 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 5.4, 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 5.3.3 *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 $\tau : (\hat{\Sigma} \cup \Sigma \cup \{i, d\})^* \to \{i, d\}^*$ 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 $R = (i\Sigma + d\hat{\Sigma})^*$. ■

Thus, we have the following corollary:

Corollary 5.3.4 *Let $\mathcal{C}$ be a cone. Let L_1, L_2, T be languages such that two are regular and the third is in $\mathcal{C}$. Then $L_1 \rightsquigarrow_T L_2 \in \mathcal{C}$.*

Note that the closure of cones under quotient with regular sets [68, Thm. 11.3] is a specific instance of Corollary 5.3.4. Lemma 5.3.1 can also be proven by appealing to Theorem 5.3.3. We also note that the CFLs are a cone, thus we have the following corollary (a direct construction is also possible):

Corollary 5.3.5 *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 5.3.5 are not met, the result might not hold:

Theorem 5.3.6 *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 $L_1 \rightsquigarrow_T L_2$ is 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 [52, Thm. 3.4]) that the CFLs are not closed under right quotient (given by the set of trajectories $T = i^*d^*$). The languages described by Ginsburg and Spanier which witness this non-closure are linear CFLs.

(b) Let $\Sigma = \{a, b, c, \#\}$ and define $L_1, L_2 \subseteq \Sigma^*$ and $T \subseteq \{i, d\}^*$ by

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

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 CSLs are not a cone, since it is known that they are not closed under arbitrary morphism (see, e.g., Mateescu and Salomaa [148, Thm. 2.12] for the closure properties of the CSLs). Thus, Corollary 5.3.4 does not apply to the CSLs. In fact, it is also known that the CSLs are not closed under (left or right) quotient with regular languages.

5.3.1 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.* [61] 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, please consult Pin [164]. 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 [164, p. 17]):

Theorem 5.3.7 *A language is regular if and only if 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 $\varphi_j : \Sigma^* \to M_j$ 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.* [61], 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 $A, B \in \mathcal{P}(M_1 \times M_2 \times M_T)$, and the product of elements of $M_1 \times M_2 \times M_T$ is defined component-wise.

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 $y \in \Sigma^*$,

$$\varphi(y)D = \{[\varphi_1(\alpha), \varphi_2(\beta), \varphi_T(t)] \ : \ y \in \alpha \rightsquigarrow_t \beta, \alpha, \beta \in \Sigma^*, t \in \{i, d\}^*\}. \tag{5.1}$$

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 (5.1), we have that

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

This establishes the following result:

Lemma 5.3.8 *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 5.3.8 gives another proof of Lemma 5.3.1.

5.3.2 Equivalence of Trajectories

We briefly note that two sets of trajectories over $\{i, d\}$ define the same deletion 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 5.3.9 *Let $T_1, T_2 \subseteq \{i, d\}^*$. Then T_1 and T_2 are equivalent if and only if $T_1 = T_2$.*

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. ■

Thus, the decidability of the equivalence problem for $T_1, T_2 \subseteq \{i, d\}^*$ is well known. For instance, it is decidable whether T_1, T_2 are equivalent if, e.g., T_1, T_2 are DCFLs, but undecidable if T_1 is regular and T_2 is an arbitrary CFL.

5.4 Regularity-Preserving Sets of Trajectories

Consider the following result of Mateescu *et al.* [147, 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 $\mathcal{H}$, which contains non-regular languages, such that for all regular languages R_1, R_2, and for all $H \in \mathcal{H}$, $R_1 \rightsquigarrow_H R_2$ is regular. We also characterize all $H \subseteq i^*d^*$ which preserve regularity (i.e., such that $R_1 \rightsquigarrow_H R_2$ is regular for all regular languages R_1, R_2), 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.

We remark that $R_1 \rightsquigarrow_H R_2$ is similar to proportional removals studied by Stearns and Hartmanis [189], Amar and Putzolu [4, 5] Seiferas and McNaughton [180], Kosaraju [120, 121, 122], Kozen [123], Zhang [205], the author [35], Berstel *et al.* [17], 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, $\frac{1}{2}(L) = L \rightsquigarrow_H \Sigma^*$. 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.

Recall that a *binary relation* r on a set S is any subset of S^2. Call a binary 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.[1]. Then, the binary relations r such that $P(\cdot, r)$ preserves regularity are precisely the u.p.-preserving relations [180].

We note the inclusion

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

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

Proof. Assume that $\rightsquigarrow_{H_r}$ 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 [180].

[1] Recall that u.p. (ultimately periodic) was defined in Section 2.1.

For the reverse implication, we modify the construction of Seiferas and McNaughton [180, Thm. 1]. Let L_1, L_2 be regular languages. 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,

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

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 [205, Sect. 3]). However, H_e is not CF. The set H_e is, however, a linear conjunctive language (see Okhotin [160] 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 [205, Thm. 5.1]). However, H_f is not a CFL, nor a linear conjunctive language [160].

Thus, there are non-CF trajectories which preserve regularity. Kozen states that there are even H_r which preserve regularity but are "highly noncomputable" [123, p. 3].

We can extend the class of non-regular sets of trajectories T such that $L_1 \rightsquigarrow_T L_2$ is regular for all regular languages L_1, L_2 by considering T such that $T \subseteq (d^* i^*)^m d^*$ for some $m \geq 1$ (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). To consider such non-regular T, it will be advantageous to adopt the notations of Zhang [205] 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,$\nabla_a(q_1, q_2) = 1$ if and only if $\delta(q_1, a) = q_2$. 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, $\nabla(q_1, q_2) = 1$ if and only if 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 $i \geq 0$, $\nabla^i(q_1, q_2) = 1$ if and only if 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 $I_{Q'}(q) = 1$ if and only if $q \in Q'$. 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 $I_{Q_1} \cdot \nabla^i \cdot I^t_{Q_2} = 1$ if and only if 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* [205], 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 $n \geq e$,

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

The functions $n!$ and 2^n are known to be m.u.p. [205].

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,

$R_1 \rightsquigarrow_T R_2$ 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 $1 \leq j \leq m$. Define $X_\ell : \mathbb{N}^m \to \mathbb{N}$ for $1 \leq \ell \leq m+1$ by

$$X_\ell(n_1, n_2, \ldots, n_m) = \sum_{j=1}^{m} f_\ell^{(j)}(n_j). \tag{5.2}$$

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

$$T = \{\prod_{j=1}^{m} (d^{X_j(\vec{n})} i^{n_j}) d^{X_{m+1}(\vec{n})} \ : \ \vec{n} = (n_1, \ldots, n_m) \in \mathbb{N}^m\}. \tag{5.3}$$

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 5.4.2 *Let $m \geq 1$, and $f_\ell^{(j)}$ be m.u.p. for $1 \leq \ell \leq m+1$ and $1 \leq j \leq m$. Let $T \subseteq (d^* i^*)^m d^*$ be defined by (5.2) and (5.3). Then for all regular languages R_1, R_2, the language $R_1 \rightsquigarrow_T R_2$ is regular.*

In this section only, let $\mathbf{m} = \{0, 1, 2, 3, \ldots, m\}$ for any $m \geq 1$.

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 $\delta_0([q_1, q_2], a) = (\delta_1[q_1, a], \delta_2[q_2, a])$ 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 $1 \leq j \leq m$ and $1 \leq \ell \leq m+1$, let $e_\ell^{(j)}$ and $p_\ell^{(j)}$ be natural numbers such that $\nabla^{f_\ell^{(j)}(n)} = \nabla^{f_\ell^{(j)}(n+p_\ell^{(j)})}$ for all $n \geq e_\ell^{(j)}$.

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

$$\mathfrak{M}(j, \ell) = \{\nabla^{f_\ell^{(j)}(i)} \ : \ 0 \leq i \leq g_\ell^{(j)}\} \times \mathbf{g}_\ell^{(\mathbf{j})}.$$

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. It is well known that multiple start states do not affect the regularity of the language accepted (see, e.g., Yu [201, p. 54]); our presentation is chosen for ease of description.

We now proceed with defining M. 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, they will not change through the course of the computation.

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

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

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 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 us 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 [205] and denote by $\langle c\rangle_a^b$ the quantity

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

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

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

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

Then we define δ by

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

Note that, though the definition of δ is complicated, its action is straight-forward. 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 5.1.

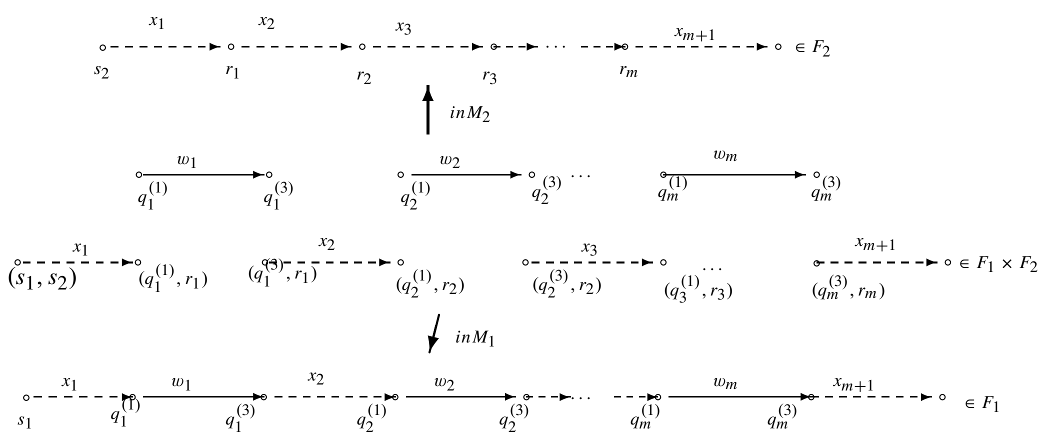

Figure 5.1: Construction of the words in M_1 and M_2 from the action of M.

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. ■

As an example, consider $m = 1$ and let $f_1^{(1)}, f_2^{(2)}$ both be the identity function. Then the conditions of Theorem 5.4.2 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 [68, 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 (5.3) 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 5.4.2, and is omitted. With this extension, we can also consider a class of examples given by Amar and Putzolu [5], 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 [5] by means of *even linear grammars*.

Pin and Sakarovitch use a very general and elegant method to prove that certain operations preserve regularity [165]. This method can be used to prove that certain operations which can be modeled by trajectories preserve regularity; it is not known whether the methods developed here can be extended to cover theses cases. For example, the let T_ζ be given by

$$T_\zeta = \{d^{nk} i^k d^{nk} \ : \ k, n \geq 0, 2n+1 \text{ is prime}\}.$$

Then Pin and Sakarovitch prove that $L \rightsquigarrow_{T_\zeta} \Sigma^*$ is regular for all regular languages L [165, p. 292][2].

[2]Note that the definition of T_ζ given here matches that given by Pin and Sakarovitch [165]. In a preliminary version [166], a different deletion operation is defined which *can* be modeled by a set of trajectories to which Theorem 5.5.1 below can be applied.

5.5 i-Regularity

Recall that a language $L \subseteq \Sigma^*$ is *bounded* if there exist $w_1, w_2, \dots, w_n \in \Sigma^*$ such that $L \subseteq w_1^* w_2^* \cdots w_n^*$. We say that L is *letter-bounded*[3] if $w_i \in \Sigma$ for all $1 \leq i \leq n$.

We now define a class of letter-bounded sets of trajectories, called i*-regular* sets of trajectories, which will have strong closure properties. In particular, we can delete, along an i-regular set of letter-bounded trajectories, any language from a regular language and the resulting language will be regular. This will allow us in Section 7.3 to give positive decidability results for the related shuffle decomposition problem.

Let Δ_m be the alphabet $\Delta_m = \{\#_1, \#_2, \dots, \#_m\}$ for any $m \geq 1$. We define a class of regular substitutions from $(d + \Delta_m)^*$ to $2^{(i+d)^*}$, denoted $\mathfrak{S}_m$, as follows: a regular substitution $\varphi : (d + \Delta_m)^* \to 2^{(i+d)^*}$ is in $\mathfrak{S}_m$ if both

(a) $\varphi(d) = \{d\}$; and

(b) for all $1 \leq j \leq m$, there exist $a_j, b_j \in \mathbb{N}$ such that $\varphi(\#_j) = i^{a_j}(i^{b_j})^*$.

For all $m \geq 1$, we also define a class of languages over the alphabet $d + \Delta_m$, denoted $\mathfrak{T}_m$, as the set of all languages $T \subseteq \#_1 d^* \#_2 d^* \cdots \#_{m-1} d^* \#_m$. Define the class of trajectories $\mathfrak{I}$ as follows:

$$\mathfrak{I} = \{T \subseteq \{i, d\}^* \ : \ \exists m \geq 1, T_m \in \mathfrak{T}_m, \varphi \in \mathfrak{S}_m \text{ such that } T = \varphi(T_m)\}.$$

If $T \in \mathfrak{I}$, we say that T is i*-regular*. As we shall see, the condition that T be i-regular is sufficient for showing that $R \rightsquigarrow_T L$ is regular for all regular languages R and all languages L.

Theorem 5.5.1 *Let $T \in \mathfrak{I}$. Then for all regular languages R and all languages L, $R \rightsquigarrow_T L$ is a regular language.*

Proof. Let $T \in \mathfrak{I}$. Let $m \geq 1$, $T' \in \mathfrak{T}_m$ and $\varphi \in \mathfrak{S}_m$ be such that $T = \varphi(T')$. Then we define $K(T) \subseteq \mathbb{N}^{m-1}$ as

$$K(T) = \{(j_1, \dots, j_{m-1}) \ : \ \#_1 d^{j_1} \#_2 d^{j_2} \cdots \#_{m-1} d^{j_{m-1}} \#_m \in T'\}.$$

[3] The term *strictly bounded* is sometimes used for this situation, e.g, Dassow *et al.* [32]. However, other sources, e.g., Harju and Karhumäki [60] and Mateescu *et al.* [150] use the same term differently.

Let a_j, b_j be defined so that $\varphi(\#_j) = i^{a_j}(i^{b_j})^*$ for all $1 \le j \le m$. Let $I_j = \{a_j + nb_j \ : \ n \ge 0\}$ for all $1 \le j \le m$.

Let R be regular and L be arbitrary. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting R. For all $q_j, q_k \in Q$, let $R(q_j, q_k) = L((Q, \Sigma, \delta, q_j, \{q_k\}))$. Note that

$$R(q_j, q_k) = \{w \in \Sigma^* \ : \ q_k \in \delta(q_j, w)\}.$$

For $I \subseteq \mathbb{N}$, let $R'_I(q_j, q_k) = R(q_j, q_k) \cap \{x \ : \ |x| \in I\}$.

We now define the set $Q_R(T, L) \subseteq Q^{2m-2}$:

$$\begin{aligned} Q_R(T, L) = \{&(q_1, q_2, \ldots, q_{2m-2}) \in Q^{2m-2} \\ &: \ \exists (k_j)_{j=1}^{m-1} \in K(T) \text{ such that } L \cap \prod_{\ell=1}^{m-1} R'_{\{k_\ell\}}(q_{2\ell-1}, q_{2\ell}) \neq \emptyset\}. \end{aligned} \tag{5.4}$$

We claim that

$$R \rightsquigarrow_T L = \bigcup_{\substack{(q_j)_{j=1}^{2m-2} \in Q_R(T,L) \\ q_f \in F}} \left(\prod_{\ell=1}^{m-1} R'_{I_\ell}(q_{2(\ell-1)}, q_{2\ell-1}) \right) \cdot R'_{I_m}(q_{2m-2}, q_f). \tag{5.5}$$

Let $x \in R \rightsquigarrow_T L$. Then we can write $x = x_1 x_2 \cdots x_m$ such that there exists some $z = z_1 z_2 \cdots z_{m-1} \in L$ such that $y = x_1 z_1 x_2 z_2 \cdots x_{m-1} z_{m-1} x_m \in R$. Further, by the conditions on T, $(|z_j|)_{j=1}^{m-1} \in K(T)$ and $|x_j| \in I_j$ for all $1 \le j \le m$. We let $q \overset{x}{\vdash} q'$ denote the fact that $\delta(q, x) = q'$ in M. As $y \in R$, there are some $q_1, q_2, \ldots, q_{2m-2}, q_f \in Q$ such that

$$q_0 \overset{x_1}{\vdash} q_1 \overset{z_1}{\vdash} q_2 \overset{x_2}{\vdash} \cdots \overset{x_{m-1}}{\vdash} q_{2m-3} \overset{z_{m-1}}{\vdash} q_{2m-2} \overset{x_m}{\vdash} q_f$$

and $q_f \in F$. Then $z_j \in R'_{\{|z_j|\}}(q_{2j-1}, q_{2j})$ for all $1 \le j \le m-1$, $x_j \in R'_{I_j}(q_{2(j-1)}, q_{2j-1})$ for all $1 \le j \le m-1$ and $x_m \in R'_{I_m}(q_{2m-2}, q_f)$. Further, note that

$$z \in L \cap \prod_{\ell=1}^{m-1} R'_{\{|z_\ell|\}}(q_{2\ell-1}, q_{2\ell}).$$

We conclude that $(q_1, q_2, \ldots, q_{2m-2}) \in Q_R(T, L)$, as $(|z_j|)_{j=1}^{m-1} \in K(T)$, and thus x is contained in the right-hand side of (5.5).

For the reverse inclusion, let $(q_1, \ldots, q_{2m-2}) \in Q_R(T, L)$ and $q_f \in F$. Let $(k_1, \ldots, k_{m-1}) \in K(T)$ be a $(m-1)$-tuple which witnesses $(q_1, q_2, \ldots, q_{2m-2})$'s membership in $Q_R(T, L)$. Then we show that $(\prod_{\ell=1}^{m-1} R'_{I_\ell}(q_{2(\ell-1)}, q_{2\ell}))R'_{I_m}(q_{2m-2}, q_f) \subseteq R \rightsquigarrow_T L$.

Let $z_j \in R'_{\{k_j\}}(q_{2j-1}, q_{2j})$ for all $1 \leq j \leq m-1$ be such that $z = z_1 \cdots z_{m-1} \in L$. Such z_j exist by definition of $Q_R(T, L)$. Let $x_j \in R'_{I_j}(q_{2(j-1)}, q_{2j-1})$ for all $1 \leq j \leq m-1$, and $x_m \in R'_{I_m}(q_{2m-2}, q_f)$ be arbitrary. Then

$$q_0 \overset{x_1}{\vdash} q_1 \overset{z_1}{\vdash} q_2 \overset{x_2}{\vdash} \cdots \overset{x_{m-1}}{\vdash} q_{2m-3} \overset{z_{m-1}}{\vdash} q_{2m-2} \overset{x_m}{\vdash} q_f.$$

Thus, $y = x_1 z_1 \cdots x_{m-1} z_{m-1} x_m \in R$. Further, the length considerations are met by definition of I_j and $(k_1, k_2, \ldots, k_{m-1}) \in K(T)$. Thus $x \in y \rightsquigarrow_T z \subseteq R \rightsquigarrow_T L$.

Thus, since $Q_R(T, L)$ is finite, $R \rightsquigarrow_T L$ is a finite union of regular languages, and thus is regular. ■

Corollary 5.5.2 *Let $T \subseteq \{i, d\}^*$ be a finite union of i-regular sets of trajectories. Then for all regular languages R and all languages L, the language $R \rightsquigarrow_T L$ is regular.*

We note that if T is not i-regular, it may define an operation which does not preserve regularity in the sense of Theorem 5.5.1. In particular, from the proof of Theorem 5.3.2, we have that if $T = (di)^*$,

$$(a^2)^*(b^2)^* \rightsquigarrow_T \{a^n b^n \ : \ n \geq 0\} = \{a^n b^n \ : \ n \geq 0\},$$

a non-regular CFL. For $T = (i + d)^*$, we have that

$$\big((ab)^* \#(ab)^* \rightsquigarrow_T \{a^n \# b^n \ : \ n \geq 0\}\big) \cap b^* a^* = \{b^n a^n \ : \ n \geq 0\}.$$

Further, if T is letter-bounded but not i-regular, then T may not preserve regularity. Again, from the proof of Theorem 5.3.2, we have that if $T = \{i^n d i^n \ : \ n \geq 0\}$. Then $a^* \# b^* \rightsquigarrow_T \{\#\} = \{a^n b^n \ : \ n \geq 0\}$. Note that in this case, the language $\{\#\}$ is a singleton. We also have that there is a non-i-regular set of trajectories,

$$T = \{i^n d^n i^n \ : \ n \geq 0\},$$

and a regular language R such that $R \rightsquigarrow_T \Sigma^*$ is not a regular language. In particular, we have the following example. Let $\Sigma = \{a, b, c\}$ and $R = a^*bc^*$. Then $R \rightsquigarrow_T \Sigma^*$ is not a regular language, as $(R \rightsquigarrow_T \Sigma^*) \cap a^*c^* = \{a^n c^n \ : \ n \geq 0\}$.

As an example of Theorem 5.5.1, consider $T = \{d^n i^m d^n \ : \ n, m \geq 0\}$. It is easily verified that $T \in \mathfrak{I}$ (consider $T' = \{\#_1 d^n \#_2 d^n \#_3 \ : \ n \geq 0\}$, and φ defined by $\varphi(\#_1) = \varphi(\#_3) = \{\epsilon\}$ and $\varphi(\#_2) = i^*$). Thus, the language $R \rightsquigarrow_T L$ is regular for all regular languages R and all languages L. For any language $L \subseteq \Sigma^*$, define $sq(L) = \{x^2 \ : \ x \in L\}$. Consider then that

$$R \rightsquigarrow_T sq(L) = \{w \ : vwv \in R, v \in L\}.$$

This precisely defines the *middle-quotient* operation, which has been investigated by Meduna [153] for linear CFLs. Let $R \mid L$ denote the middle quotient of R by L, i.e., $R \mid L = R \rightsquigarrow_T sq(L)$. Thus, we can immediately conclude the following result, which was not considered by Meduna:

Theorem 5.5.3 *Given a regular language R and arbitrary language L, the language $R|L$ is regular.*

5.6 Filtering and Deletion along Trajectories

Recently, Berstel *et al.* [17] introduced the concept of filtering. Here we examine the notion of filtering, and show that it is a particular case of deletion along trajectories.

Given a sequence $s \subseteq \mathbb{N}$, and a word $w \in \Sigma^*$ with $w = w_1 \cdots w_n$, $w_i \in \Sigma$, the filtering of w by s is given by $w[s] = w_{s_0} w_{s_1} \cdots w_{s_k}$ where k is such that $s_k \leq n < s_{k+1}$. For example, if $s = (1, 2, 4, 7)$, then $abcacb[s] = aba$. Filtering is extended monotonically to languages.

For every $s \subseteq \mathbb{N}$, let $\omega_s \ : \ \mathbb{N} \to \{i, d\}$ be given by $\omega_s(j) = i$ if $j \in s$ and $\omega_s(j) = d$ otherwise[4]. Let $T_s \subseteq \{0, 1\}^*$ be defined by

$$T_s = \{\prod_{j=0}^{n} \omega_s(j) \ : \ n \geq 0\}.$$

Then we clearly have that

$$L[s] = L \rightsquigarrow_{T_s} \Sigma^*,$$

[4]That is, ω_s is the characteristic ω-word of s over $\{i, d\}$.

for all sequences $s \subseteq \mathbb{N}$. Note that for all $s \subseteq \mathbb{N}$, T_s is *prefix-closed* (i.e., if $t_1 \in T_s$ and t_2 is a prefix of t_1, then $t_2 \in T_s$).

For all sequences $s = (s_j)_{j\geq 1} \subseteq \mathbb{N}$, let $\partial s = ((\partial s)_j)_{j\geq 1}$ be defined by $(\partial s)_j = s_{j+1} - s_j$ for $j \geq 1$. The sequence ∂s is called the differential sequence of s. A sequence $s \subseteq \mathbb{N}$ is said to be *residually ultimately periodic* if for each finite monoid F and each monoid morphism $\varphi : \mathbb{N} \rightarrow F$, $\varphi(s)$ is ultimately periodic.

Berstel *et al.* [17] characterize those sequences $s \subseteq \mathbb{N}$ which preserve regularity. In particular, a sequence s preserves regularity if and only if it is *differentially residually ultimately periodic*, i.e., the sequence ∂s is residually ultimately periodic.

5.7 Splicing on Routes

Splicing on routes was introduced by Mateescu [144] to model generalizations of the crossover splicing operation (see Mateescu [144] for a definition of the crossover splicing operation). Crossover splicing simulates the manner in which two DNA strands may be spliced together at multiple locations to form several new strands, see Mateescu for a discussion [144]. Splicing on routes has also been used to model dialogue in natural languages [12].

Splicing on routes generalizes the crossover splicing operation by specifying a set T of routes which restricts the way in which splicing can occur. The result is that specific sets of routes can simulate not only the crossover operation, but also such operations on DNA such as the *simple splicing* and the *equal-length crossover* operations (see Mateescu for details and definitions of these operations [144]). Splicing on routes is also a generalization of the shuffle on trajectories operation.

In this section, we consider the simulation of splicing on a route by shuffle and deletion along trajectories. We show that there exist three fixed weak codings π_1, π_2, π_3 such that for all routes t, we can simulate the splicing on t of two words w_1, w_2 by a fixed combination of the shuffle and deletion of the same languages w_1, w_2 along the trajectories $\pi_1(t), \pi_2(t), \pi_3(t)$. As a corollary, it is shown that every unary operation defined by splicing on routes can also be performed by a deletion along trajectories.

We define the concept of *splicing on routes*, and note the difference between deletion along trajectories from splicing on routes, which allows discarding letters from either input word. In particular, a *route* is a word t specified over the alphabet $\{0, \overline{0}, 1, \overline{1}\}$, where, informally, 0, 1 means insert the letter from the appropriate word, and $\overline{0}, \overline{1}$ means discard that letter and continue.

Formally, let $x, y \in \Sigma^*$ and $t \in \{0, \overline{0}, 1, \overline{1}\}^*$. We define the splicing of x and y, denoted $x \bowtie_t y$, recursively as follows: if $x = ax'$, $y = by'$ $(a, b \in \Sigma)$ and $t = ct'$ $(c \in \{0, \overline{0}, 1, \overline{1}\})$, then

$$x \bowtie_{ct'} y = \begin{cases} a(x' \bowtie_{t'} y) & \text{if } c = 0; \\ (x' \bowtie_{t'} y) & \text{if } c = \overline{0}; \\ b(x \bowtie_{t'} y') & \text{if } c = 1; \\ (x \bowtie_{t'} y') & \text{if } c = \overline{1}. \end{cases}$$

If $x = ax'$ and $t = ct'$, where $a \in \Sigma$ and $c \in \{0, \overline{0}, 1, \overline{1}\}$, then

$$x \bowtie_{ct'} \epsilon = \begin{cases} a(x' \bowtie_{t'} \epsilon) & \text{if } c = 0; \\ (x' \bowtie_{t'} \epsilon) & \text{if } c = \overline{0}; \\ \emptyset & \text{otherwise.} \end{cases}$$

If $y = by'$ and $t = ct'$, where $a \in \Sigma$ and $c \in \{0, \overline{0}, 1, \overline{1}\}$, then

$$\epsilon \bowtie_{ct'} y = \begin{cases} b(\epsilon \bowtie_{t'} y') & \text{if } c = 1; \\ (\epsilon \bowtie_{t'} y') & \text{if } c = \overline{1}; \\ \emptyset & \text{otherwise.} \end{cases}$$

We have $x \bowtie_\epsilon y = \epsilon$ if $\{x, y\} \neq \{\epsilon\}$. Finally, we set $\epsilon \bowtie_t \epsilon = \epsilon$ if $t = \epsilon$ and $\emptyset$ otherwise. We extend $\bowtie_t$ to sets of trajectories and languages as expected:

$$\begin{aligned} x \bowtie_T y &= \bigcup_{t \in T} x \bowtie_t y \quad \forall T \subseteq \{0, \overline{0}, 1, \overline{1}\}^*, x, y \in \Sigma^*; \\ L_1 \bowtie_T L_2 &= \bigcup_{\substack{x \in L_1 \\ y \in L_2}} x \bowtie_T y. \end{aligned}$$

For example, if $x = abc$, $y = cbc$ and $T = \{010011, 0\overline{1}0\overline{0}11\}$, then $x \bowtie_T y = \{acbcbc, abbc\}$.

We now demonstrate that splicing on routes can be simulated by a combination of shuffle on trajectories and deletion along trajectories.

Theorem 5.7.1 *There exist weak codings $\pi_1, \pi_2 : \{0, 1, \overline{0}, \overline{1}\}^* \to \{i, d\}^*$ and a weak coding $\pi_3 : \{0, 1, \overline{0}, \overline{1}\}^* \to \{0, 1\}^*$ such that for all $t \in \{0, \overline{0}, 1, \overline{1}\}^*$, and for all $x, y \in \Sigma^*$, we have*

$$x \bowtie_t y = (x \rightsquigarrow_{\pi_1(t)} \Sigma^*) \sqcup\!\sqcup_{\pi_3(t)} (y \rightsquigarrow_{\pi_2(t)} \Sigma^*).$$

Proof. Let $\pi_1, \pi_2 : \{0, \overline{0}, 1, \overline{1}\}^* \to \{i, d\}^*$ and $\pi_3 : \{0, \overline{0}, 1, \overline{1}\} \to \{0, 1\}^*$ be given by

$$\begin{array}{llllllllllll}
\pi_1(0) & = & i; & \pi_1(\overline{0}) & = & d; & \pi_1(1) & = & \epsilon; & \pi_1(\overline{1}) & = & \epsilon; \\
\pi_2(0) & = & \epsilon; & \pi_2(\overline{0}) & = & \epsilon; & \pi_2(1) & = & i; & \pi_2(\overline{1}) & = & d; \\
\pi_3(0) & = & 0; & \pi_3(\overline{0}) & = & \epsilon; & \pi_3(1) & = & 1; & \pi_3(\overline{1}) & = & \epsilon.
\end{array}$$

We first show the left-to-right inclusion. Let $z \in x \bowtie_t y$. The result is by induction on $|t|$. If $|t| = 0$, then $x = y = z = \epsilon$. Thus, we can easily verify that $z \in (\epsilon \rightsquigarrow_\epsilon \epsilon) \sqcup\!\sqcup_\epsilon (\epsilon \rightsquigarrow_\epsilon \epsilon)$.

Let $|t| > 0$. Then $t = ct'$ for $c \in \{0, \overline{0}, 1, \overline{1}\}$. We prove only the case where $c = 0$ and $c = \overline{0}$. The other two cases are similar and are left to the reader.

(a) $c = 0$. Then $x = ax'$ and $z \in a(x' \bowtie_{t'} y)$ for some $x' \in \Sigma^*$. Thus, $z = az'$ for some $z' \in (x' \bowtie_{t'} y)$. By induction, $z' \in (x' \rightsquigarrow_{\pi_1(t')} \Sigma^*) \sqcup\!\sqcup_{\pi_3(t')} (y \rightsquigarrow_{\pi_2(t')} \Sigma^*)$. Let $u \in x' \rightsquigarrow_{\pi_1(t')} \Sigma^*$ and $v \in y \rightsquigarrow_{\pi_2(t')} \Sigma^*$ be such that $z' \in u \sqcup\!\sqcup_{\pi_3(t')} v$.

Note that $\pi_1(t) = i\pi_1(t')$. Thus by definition of $\rightsquigarrow_T$, $au \in x \rightsquigarrow_{\pi_1(t)} \Sigma^*$. Similarly, as $\pi_2(t) = \pi_2(t')$, $v \in y \rightsquigarrow_{\pi_2(t)} \Sigma^*$. Finally $\pi_3(t) = 0\pi_3(t')$. Thus, $au \sqcup\!\sqcup_{\pi_3(t)} v = a(u \sqcup\!\sqcup_{\pi_3(t')} v) \ni az' = z$. Thus, the result holds for $c = 0$.

(b) $c = \overline{0}$. Then $x = ax'$ and $z \in (x' \bowtie_{t'} y)$ for some $x' \in \Sigma^*$. Thus, by induction $z \in (x' \rightsquigarrow_{\pi_1(t')} \Sigma^*) \sqcup\!\sqcup_{\pi_3(t')} (y \rightsquigarrow_{\pi_2(t')} \Sigma^*)$. Let $u \in x' \rightsquigarrow_{\pi_1(t')} \Sigma^*$ and $v \in y \rightsquigarrow_{\pi_2(t')} \Sigma^*$ be such that $z \in u \sqcup\!\sqcup_{\pi_3(t')} v$.

Note in this case that $\pi_1(t) = d\pi_1(t')$. Thus, $u \in x \rightsquigarrow_{\pi_1(t)} \Sigma^*$. Similarly, as $\pi_2(t) = \pi_2(t')$, $v \in y \rightsquigarrow_{\pi_2(t)} \Sigma^*$. Finally, $\pi_3(t) = \pi_3(t')$. Thus, $u \sqcup\!\sqcup_{\pi_3(t)} v = u \sqcup\!\sqcup_{\pi_3(t')} v \ni z$. Thus, the result holds for $c = \overline{0}$.

We now prove the reverse inclusion. Let $z \in (x \rightsquigarrow_{\pi_1(t)} \Sigma^*) \sqcup\!\sqcup_{\pi_3(t)} (y \rightsquigarrow_{\pi_2(t)} \Sigma^*)$. We show the result by induction on t. For $|t| = 0$, $t = \epsilon$. Thus $\pi_1(t) = \pi_2(t) = \pi_3(t) = \epsilon$. By definition of $\sqcup\!\sqcup_t$, $L_1 \sqcup\!\sqcup_\epsilon L_2$ is non-empty if and only if $\epsilon \in L_1 \cap L_2$, which implies $z = \epsilon$. Thus, $\epsilon \in (x \rightsquigarrow_\epsilon \Sigma^*)$, and similarly for y in place of x. By definition of $\rightsquigarrow_t$, this implies that $x = y = \epsilon$. Thus, $z \in x \bowtie_t y$, by definition. The inclusion is proven for $|t| = 0$.

Let $|t| > 0$. Thus, there is some $c \in \{0, \overline{0}, 1, \overline{1}\}$, and $t' \in \{0, \overline{0}, 1, \overline{1}\}^*$ such that $t = ct'$. We distinguish between four cases, for each choice of c in $\{0, \overline{0}, 1, \overline{1}\}$, however, we only prove the cases $c = 0$ and $c = \overline{0}$. The other two cases are very similar, and are left to the reader.

(a) $c = 0$. Note that $\pi_1(t) = i\pi_1(t')$, $\pi_2(t) = \pi_2(t')$ and $\pi_3(t) = 0\pi_3(t')$. Let $u, v \in \Sigma^*$ be words such that $u \in x \rightsquigarrow_{\pi_1(t)} \Sigma^*$, $v \in y \rightsquigarrow_{\pi_2(t)} \Sigma^*$ and $z \in u \sqcup\!\sqcup_{\pi_3(t)} v$.

As $\pi_3(t) = 0\pi_3(t')$, we have, by definition of $\sqcup\!\sqcup_t$, that $u = au'$, $z = az'$ and $z' \in u' \sqcup\!\sqcup_{\pi_3(t')} v$ for some $a \in \Sigma$ and $u', z' \in \Sigma^*$. Now, as $au' \in x \rightsquigarrow_{i\pi_1(t')} \Sigma^*$, there exists $x' \in \Sigma^*$ such that $x = ax'$ and $u' \in x' \rightsquigarrow_{\pi_1(t')} \Sigma^*$. Also, note that $v \in y \rightsquigarrow_{\pi_2(t')} \Sigma^*$. Thus, combining these yields that

$$z' \in (x' \rightsquigarrow_{\pi_1(t')} \Sigma^*) \sqcup\!\sqcup_{\pi_3(t')} (y \rightsquigarrow_{\pi_2(t')} \Sigma^*).$$

By induction, $z' \in x' \bowtie_{t'} y$. Thus,

$$z = az' \in a(x' \bowtie_{t'} y) = ax' \bowtie_{0t'} y = x \bowtie_t y.$$

Thus, the inclusion is proven.

(b) $c = \overline{0}$. Then $\pi_1(t) = d\pi_1(t')$, $\pi_2(t) = \pi_2(t')$ and $\pi_3(t) = \pi_3(t')$. Let $u, v \in \Sigma^*$ be such that $u \in x \rightsquigarrow_{\pi_1(t)} \Sigma^*$, $v \in y \rightsquigarrow_{\pi_2(t)} \Sigma^*$ and $z \in u \sqcup\!\sqcup_{\pi_3(t)} v$.

As $u \in x \rightsquigarrow_{\pi_1(t)} \Sigma^*$, let $u_0 \in \Sigma^*$ be such that $u \in x \rightsquigarrow_{\pi_1(t)} u_0$. As $\pi_1(t) = d\pi_1(t')$, there are some $b \in \Sigma$, $x', u_0' \in \Sigma^*$ such that $x = bx'$, $u_0 = bu_0'$ and $u \in x' \rightsquigarrow_{\pi_1(t')} u_0'$. Thus, $u \in x' \rightsquigarrow_{\pi_1(t')} \Sigma^*$. Note that $v \in y \rightsquigarrow_{\pi_2(t')} \Sigma^*$. Thus, $z \in u \sqcup\!\sqcup_{\pi_3(t')} v \subseteq (x' \rightsquigarrow_{\pi_1(t')} \Sigma^*) \sqcup\!\sqcup_{\pi_3(t')} (y \rightsquigarrow_{\pi_2(t')} \Sigma^*)$. By induction, $z \in x' \bowtie_{t'} y$. Thus, we can see that $(bx' \bowtie_t y) = x' \bowtie_{t'} y \ni z$. This proves the inclusion.

The result is now proven. ■

Corollary 5.7.2 *There exist weak codings* $\pi_1, \pi_2 : \{0, 1, \overline{0}, \overline{1}\}^* \to \{i, d\}^*$ *and* $\pi_3 : \{0, 1, \overline{0}, \overline{1}\}^* \to \{0, 1\}^*$ *such that for all* $T \subseteq \{0, \overline{0}, 1, \overline{1}\}^*$ *and* $L_1, L_2 \subseteq \Sigma^*$,

$$L_1 \bowtie_T L_2 = \bigcup_{t \in T} (L_1 \rightsquigarrow_{\pi_1(t)} \Sigma^*) ⧢_{\pi_3(t)} (L_2 \rightsquigarrow_{\pi_2(t)} \Sigma^*).$$

Unfortunately, the identity

$$L_1 \bowtie_T L_2 = (L_1 \rightsquigarrow_{\pi_1(T)} \Sigma^*) ⧢_{\pi_3(T)} (L_2 \rightsquigarrow_{\pi_2(T)} \Sigma^*)$$

does not hold in general, even if L_1, L_2 are singletons and $|T| = 2$. For example, if $L_1 = \{ab\}$, $L_2 = \{cd\}$ and $T = \{\overline{0}01\overline{1}, 0\overline{0}\overline{1}1\}$, then

$$\begin{aligned} L_1 \bowtie_T L_2 &= \{bc, ad\}; \\ (L_1 \rightsquigarrow_{\pi_1(T)} \Sigma^*) ⧢_{\pi_3(T)} (L_2 \rightsquigarrow_{\pi_2(T)} \Sigma^*) &= \{ac, ad, bc, bd\}. \end{aligned}$$

However, if T is a *unary* set of routes, by which we mean that $T \subseteq \{0, \overline{0}\}^* \overline{1}^*$, then we have the following result, which is easily established:

Corollary 5.7.3 *Let* $T \subseteq \{0, \overline{0}\}^* \overline{1}^*$. *Then for all* $L \subseteq \Sigma^*$,

$$L \bowtie_T \Sigma^* = L \rightsquigarrow_{\pi_1(T)} \Sigma^*.$$

We refer the reader to Mateescu [144] for a discussion of unary operations defined by splicing on routes. As an example, consider that with $T = \{0^n \overline{0}^n : n \geq 0\} \overline{1}^*$, $L \bowtie_T \Sigma^* = \frac{1}{2}(L)$, where $\frac{1}{2}(L)$ was given in Section 5.4.

5.8 Inverse Word Operations

In this section, we show that deletion along trajectories constitutes the inverse of shuffle on trajectories, in the sense introduced by Kari [106].

We now define a word operation for our purposes. Given an alphabet Σ^*, a *word operation* is any binary function $\diamond : (\Sigma^*)^2 \to 2^{\Sigma^*}$. We usually denote a word operation as an infix operator. A word operation is extended to languages in a monotone way, as we have already seen for shuffle and deletion along trajectories: given $L_1, L_2 \subseteq \Sigma^*$,

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

Note that unlike Hsiao *et al.* [69], we do not make any assumptions about the action of $\diamond$ on ϵ as an argument.

5.8.1 Left Inverse

Given two binary word operations $\diamond, \star : (\Sigma^*)^2 \to 2^{\Sigma^*}$, we say that $\diamond$ is a *left-inverse* of $\star$ [106, Defn. 4.1] if, for all $u, v, w \in \Sigma^*$,

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

For instance, the operations of concatenation and right-quotient are left-inverses of each other, as $w = uv$ iff $u \in w/v$.

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 5.8.1 *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

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

Let $w \in \Sigma^*$ with $|w| > 0$ and assume that the result is true for all words shorter than w. Let $w = aw'$ for $a \in \Sigma$.

First, assume that $aw' \in u \sqcup\!\sqcup_t v$. 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 $w' \in u' \sqcup\!\sqcup_{t'} v$. 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 $w' \in u \sqcup\!\sqcup_{t'} v'$. 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 $w' \in u' \sqcup\!\sqcup_{t'} v$. 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 5.8.1 agrees with the observations of Kari [106, Obs. 4.7].

5.8.2 Right Inverse

Given two binary word operations $\diamond, \star : (\Sigma^*)^2 \to 2^{\Sigma^*}$, we say that $\diamond$ is a *right-inverse* [106, 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* $\diamond$ [106].

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 5.8.2 *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 defined by $sym_s(0) = 1$ and $sym_s(1) = 0$. Then it is easy to note (cf., Mateescu *et al.* [147, Rem. 4.9(i)]) that

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

Thus, using Theorem 5.8.1, 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 [106, Obs. 4.4].

We also consider 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 5.8.3 *Let $T \subseteq \{i, d\}^*$ be a set of trajectories. The operation $\rightsquigarrow_T$ has right-inverse $\rightsquigarrow_{sym_d(T)}$.*

Proof. By Theorems 5.8.2 and 5.8.1, 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 $\pi \circ \tau^{-1} \equiv sym_d$. ■

We note that Theorem 5.8.3 agrees with the observations of Kari [106, Obs. 4.4].

5.9 Conclusions

We have defined deletion along trajectories, and examined its closure properties. Deletion along trajectories is shown to be a useful generalization of the 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 deletion operations which preserve regularity.

We have also demonstrated that shuffle on trajectories and deletion along trajectories form mutual inverses of each other in the sense of Kari [106]. In Chapter 7, we will use the fact that shuffle and deletion along trajectories are mutual inverses of each other to solve language equations involving these operations. In Chapter 6, we will use the inverse characterizations to allow us to prove positive decidability results.