Chapter 4

Descriptional Complexity

4.1 Introduction

Descriptional complexity of formal languages deals with the problems of concise descriptions of languages in terms of generative or accepting devices. For instance, the *(deterministic) state complexity* of a regular language L is the minimal number of states in any deterministic finite automaton accepting L [204]. Nondeterministic state complexity of a regular language is similarly defined [48, 65, 66].

There is much interest in descriptional complexity as it relates to the efficiency of implementing operations on languages. For instance, if f is a binary operation which preserves regular languages, then research in state complexity typically seeks to express the *worst-case* state complexity of $f(L_1, L_2)$ as a function of the state complexities of L_1 and L_2. Informally, we refer to this expression for the complexity of $f(L_1, L_2)$ as *the state complexity* of f. For a survey of worst-case state complexity for finite and regular languages, see Yu [202, 203]. We note that research into *average-case* state complexity (instead of worst-case) of f has also been examined by Nicaud [155] and the author [35].

For shuffle on trajectories, Mateescu *et al.* [147] and Harju *et al.* [61] both give proofs that, given a regular set of trajectories T and regular languages L_1, L_2, the operation $L_1 \sqcup\!\sqcup_T L_2$ always yields a regular language. Thus, it is reasonable to consider the state complexity of shuffle on trajectories; this is the goal of this chapter.

It is known that each set $T \subseteq \{0, 1\}^*$ defines a unique operation $\sqcup\!\sqcup_T$ (to see this, consider that $0^* \sqcup\!\sqcup_T 1^* = T$). Therefore, the family of shuffle on trajectory operations is very complex, and in this study we only begin to address the many questions which arise from studying the state complexities of these operations. We incorporate other measures of complexity used in formal languages and automata theory, including nondeterministic state complexity and language density (for a definition of density of languages, see Section 4.3).

In particular, we establish a general upper bound, and improve it in the case when the set of trajectories T has constant density. For sets of trajectories with density one, we obtain a lower bound that is of the same order as the upper bound when the state complexity of the set of trajectories grows with respect to the state complexity of the component languages.

We also consider a result of Yu *et al.* [204] on the state complexity of the concatenation operation. We show that the state complexity of L_1L_2 can be improved in the case that L_2 can be easily accepted by a NFA. However, this is not an improvement in the worst case.

4.2 General State Complexity Bounds

Given a regular language L, define the *(deterministic) state complexity* of L, denoted $sc(L)$, by

$$sc(L) = \min\{|Q| \ : \ M = (Q, \Sigma, \delta, q_0, F) \text{ is a DFA accepting } L\}.$$

It is well known that for a regular language L, $sc(L)$ is the index of $\equiv_L$, the Myhill-Nerode congruence with respect to L. The *nondeterministic state complexity* of a regular language L is defined similarly by

$$nsc(L) = \min\{|Q| \ : \ M = (Q, \Sigma, \delta, q_0, F) \text{ is an NFA accepting } L\}.$$

Nondeterministic state complexity has recently been studied by Holzer and Kutrib [65, 66] and Ellul [48].

The following theorem [147, Thm. 5.1] states that regular sets of trajectories preserve regularity. It serves as the starting point of this chapter:

Theorem 4.2.1 *Let L_1, L_2 be regular languages over Σ^* and let $T \subseteq \{0,1\}^*$ be a regular language. Then $L_1 \sqcup\!\sqcup_T L_2$ is a regular language.*

The construction given by Mateescu *et al.* [147, Thm. 5.1] yields our most general upper bound on the state complexity of shuffle on trajectories. We state our upper bound in terms of nondeterministic state complexity:

Lemma 4.2.2 *Let L_1, L_2 be regular languages over Σ^* and $T \subseteq \{0,1\}^*$ be a regular set of trajectories. Then*

$$sc(L_1 \sqcup\!\sqcup_T L_2) \leq 2^{nsc(L_1)nsc(L_2)nsc(T)}.$$

Proof. We construct a NFA M' accepting $L_1 \sqcup\!\sqcup_T L_2$. Let $M_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$ be minimal NFAs accepting L_i for $i = 1$ and 2, and let $M_T = (Q_T, \{0,1\}, \delta_T, q_T, F_T)$ be a minimal NFA accepting T.

Let $M' = (Q, \Sigma, \delta, q_0, F)$ be an NFA with $Q = Q_1 \times Q_2 \times Q_T$, $q_0 = [q_1, q_2, q_T]$, $F = F_1 \times F_2 \times F_T$ and δ given by

$$\begin{aligned}\delta([q_i, q_j, q_k], a) &= \{[q, q_j, q'] : q \in \delta_1(q_i, a), q' \in \delta_T(q_k, 0)\} \\ &\quad \cup \{[q_i, q, q'] : q \in \delta_2(q_j, a), q' \in \delta_T(q_k, 1)\}\end{aligned}$$

for all $q_i \in Q_1, q_j \in Q_2, q_k \in Q_T$ and $a \in \Sigma$. Then it is easily verified that $L(M') = L_1 \sqcup\!\sqcup_T L_2$. Since M' is an NFA with $nsc(L_1)nsc(L_2)nsc(T)$ states, the result easily follows, since any NFA with n states can be simulated by a DFA with 2^n states. ■

Thus, we have the following interesting corollary:

Corollary 4.2.3 *Let L_1, L_2 be regular languages over Σ^* and $T \subseteq \{0,1\}^*$ be a regular set of trajectories. If*

$$sc(L_1 \sqcup\!\sqcup_T L_2) = 2^{sc(L_1)sc(L_2)sc(T)}$$

then $sc(L_i) = nsc(L_i)$ for $i = 1, 2$ and $sc(T) = nsc(T)$.

Proof. As $nsc(L) \leq sc(L)$ for all regular languages L, the result is evident. ■

Using the idea of Lemma 4.2.2, we may slightly modify a result of Yu *et al.* [204] concerning concatenation:

Theorem 4.2.4 *Let $L_1, L_2 \subseteq \Sigma^*$ be regular languages. Then*

$$sc(L_1 L_2) \leq sc(L_1) 2^{nsc(L_2)} - k 2^{nsc(L_2)-1},$$

where k is the number of final states in the minimal DFA accepting L_1.

This is not an improvement in the worst case, but it again shows that if L_1, L_2 are languages with $sc(L_1 L_2) = sc(L_1) 2^{sc(L_2)} - k 2^{sc(L_2)-1}$ then $nsc(L_2) = sc(L_2)$. This applies to the lower bound given by Yu *et al.*: Let $M_B = (\{p_0, p_1, \ldots, p_n\}, \{a, b, c\}, \delta_B, p_0, \{p_{n-1}\})$ be a DFA with δ_B given by

$$\begin{aligned} \delta_B(p_i, a) &= p_i; \\ \delta_B(p_i, b) &= p_{i+1}; \\ \delta_B(p_i, c) &= p_1; \end{aligned}$$

where the indices are taken modulo n. Then if $L = L(M_B)$, $sc(L) = nsc(L)$. Thus, the language given by the above DFA cannot be accepted by an NFA with any less states.

Also note that Theorem 4.2.4 demonstrates that there exist sets of trajectories T for which Lemma 4.2.2 is not optimal. In particular, concatenation is given by the set of trajectories $T = 0^* 1^*$, that is, $\sqcup\!\sqcup_T = \cdot$, the concatenation operator. Since $nsc(0^* 1^*) = sc(0^* 1^*) - 1 = 2$ (see Figure 4.1), Lemma 4.2.2 gives $sc(L_1 L_2) \leq 4^{nsc(L_1) nsc(L_2)}$. However, by Theorem 4.2.4, we get that

$$sc(L_1 L_2) \leq sc(L_1) 2^{nsc(L_2)} \leq 2^{nsc(L_1) + nsc(L_2)}.$$

Thus, we have the following problem:

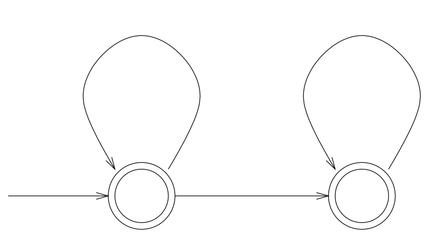

Figure 4.1: A two-state NFA accepting the set $T = 0^*1^*$ of trajectories.

Open Problem 4.2.5 *For what regular sets of trajectories $T \subseteq \{0, 1\}^*$ does the construction given by Lemma 4.2.2 give a construction which is best possible?*

Consider unrestricted shuffle, given by the set of trajectories $T = (0 + 1)^*$. The bound of Lemma 4.2.2 in this case is $2^{nsc(L_1)nsc(L_2)}$. Câmpeanu *et al.* [22] have shown that there exist languages L_1 and L_2 accepted by incomplete DFAs having, respectively, n and m states such that any incomplete DFA accepting $L_1 \sqcup\!\sqcup L_2$ has at least $2^{nm} - 1$ states. This bound is optimal for incomplete DFAs, however; for complete DFAs it gives only the lower bound $2^{(sc(L_1)-1)(sc(L_2)-1)}$. However, we regard this as near enough to our goal of Lemma 4.2.2 for our purposes, i.e., we regard $T = (0+1)^*$ as an example of a set of trajectories T satisfying Open Problem 4.2.5.

4.3 Slenderness and Trajectories

In this section, we consider the opposite question to Open Problem 4.2.5. That is, we are interested in finding $T \subseteq \{0, 1\}^*$ such that Lemma 4.2.2 is not optimal, and in fact, is a very poor bound. To define such T, we examine another descriptional complexity measure on languages, that of the *density*. Informally, the density of a language measures the number of words of each length. We find that sets of trajectories T with very small density yield operations $\sqcup\!\sqcup_T$ with small state complexity, compared to Lemma 4.2.2.

We now give the definition of the *density function* of a language $L \subseteq \Sigma^*$. For all $n \geq 0$, define $p_L : \mathbb{N} \to \mathbb{N}$ as

$$p_L(n) = |L \cap \Sigma^n|.$$

That is, $p_L(n)$ gives the number of words of length n in L. By the density of a language L, we informally mean the asymptotic behaviour of p_L. The following important result of Szilard *et al.* [190, Thm. 3] characterizes the density of regular languages:

Theorem 4.3.1 *A regular language R over Σ satisfies $p_R(n) \in O(n^k)$, $k \geq 0$ if and only if R can be represented as a finite union of regular expressions of the following form:*

$$xy_1^* z_1 \cdots y_t^* z_t$$

where $x, y_1, z_1, \cdots, y_t, z_t \in \Sigma^*$, and $0 \leq t \leq k+1$.

Call a language L *slender* if $p_L(n) \in O(1)$ [168]. If a regular language R has polynomial density $O(n^k)$, let t be the smallest integer such that $R = \cup_{i=1}^{t} x_i y_{i,1}^* z_{i,1} \cdots y_{i,k_i}^* z_{i,k_i}$, $0 \leq k_i \leq k+1$, $i = 1, \ldots, t$. Then call t the *UkL-index* of L. If $k = 0$, we call t the *USL-index* of L (languages with USL index t are called t-thin by Păun and Salomaa [168]; slender regular languages were also characterized independently by Shallit [181, Lemma 3, p. 336]).

4.3.1 Perfect Shuffle

We first consider a common example of a slender set of trajectories, that of perfect (or balanced literal) shuffle. Recall that perfect shuffle is given by the set of trajectories $T_p = (01)^*$; we denote the perfect shuffle operation by $\sqcup\!\sqcup_p$. Thus, for $x, y \in \Sigma^*$, $x = x_1 x_2 \cdots x_m$, $y = y_1 y_2 \cdots y_n$, where $x_i, y_j \in \Sigma$, the perfect shuffle of x and y is

$$x \sqcup\!\sqcup_p y = \begin{cases} x_1 y_1 x_2 y_2 \cdots x_m y_m & \text{if } m = n; \\ \emptyset & \text{otherwise.} \end{cases}$$

The following result can be obtained directly. However, we will defer the proof by stating that it is an immediate corollary of Lemma 4.3.4, which appears below in Section 4.3.2:

Lemma 4.3.2 *Let L_1, L_2 be regular languages with $sc(L_i) = n_i$ for $i = 1, 2$. Then*

$$sc(L_1 \sqcup\!\sqcup_p L_2) \leq 2n_1 n_2.$$

We can show this to be optimal for all n_1, n_2 over a two-letter alphabet.

Lemma 4.3.3 *Let $\Sigma = \{a, b\}$. Let $n_1, n_2 \geq 0$ be integers. Then there exist regular languages $L_1, L_2 \subseteq \Sigma^*$ with $sc(L_i) = n_i$ for $i = 1, 2$ such that $sc(L_1 \sqcup\!\sqcup_p L_2) = 2n_1n_2$.*

Proof. Let $L_1 = \{x \in \{a, b\}^* : |x|_a \equiv 0 \pmod{n_1}\}$ and $L_2 = \{x \in \{a, b\}^* : |x|_b \equiv 0 \pmod{n_2}\}$. It is easily verified that $sc(L_1) = n_1$ and $sc(L_2) = n_2$. We claim that $sc(L_1 \sqcup\!\sqcup_p L_2) \geq 2n_1n_2$.

We consider words of the form $a^{2i}b^j$ for $0 \leq i < n_1$ and $0 \leq j \leq 2n_2 - 1$. For any pairs $[i_1, j_1] \neq [i_2, j_2]$, we have that $a^{2i_1}b^{j_1} \not\equiv_L a^{2i_2}b^{j_2}$ (where $L = L_1 \sqcup\!\sqcup_p L_2$). To show this, we show that any two distinct words $w_1 = a^{2i_1}b^{j_1}$ and $w_2 = a^{2i_2}b^{j_2}$ can be distinguished with the word $u = a^{2(n_1-i_1)}b^{2n_2-j_1}$. We establish now that $w_1 \not\equiv_L w_2$ by showing that $w_1u \in L$ while $w_2u \notin L$.

Case (i): j_1, j_2 both odd. Let $0 \leq j_1', j_2' < n_2$ be integers such that $j_1 = 2j_1' + 1$ and $j_2 = 2j_2' + 1$.

Consider $w_1u = a^{2i_1}b^{2j_1'+1}a^{2(n_1-i_1)}b^{2(n_2-j_1')-1}$. Then $w_1u = v_1 \sqcup\!\sqcup_p v_2$ where

$$\begin{aligned} v_1 &= a^{i_1}b^{j_1'+1}a^{n_1-i_1}b^{n_2-j_1'-1}; \\ v_2 &= a^{i_1}b^{j_1'}a^{n_1-i_1}b^{n_2-j_1'}. \end{aligned}$$

Thus $|v_1|_a = n_1$ and $|v_2|_b = n_2$ and so $w_1u \in L$.

As for $w_2u = a^{2i_2}b^{2j_2'+1}a^{2(n_1-i_1)}b^{2(n_2-j_1')-1}$, we have $w_2u = v_3 \sqcup\!\sqcup_p v_4$ where

$$\begin{aligned} v_3 &= a^{i_2}b^{j_2'+1}a^{n_1-i_1}b^{n_2-j_1'-1}; \\ v_4 &= a^{i_2}b^{j_2'}a^{n_1-i_1}b^{n_2-j_1'}. \end{aligned}$$

Then note that $|v_3|_a = n_1 - i_1 + i_2$ and $|v_4|_b = n_2 - j_1' + j_2'$. Since $0 \leq i_1, i_2 < n_1$ and $0 \leq j_1', j_2' < n_2$, and under the assumptions that one of $i_1 \neq i_2$ and $j_1 \neq j_2$ is true, we have either $v_3 \notin L_1$ or $v_4 \notin L_2$. Thus, $w_2u \notin L$.

Case (ii): j_1, j_2 both even. Let $0 \leq j_1', j_2' < n_2$ be integers such that $j_1 = 2j_1'$ and $j_2 = 2j_2'$.

Consider $w_1u = a^{2i_1}b^{2j_1'}a^{2(n_1-i_1)}b^{2(n_2-j_1')}$. Again, decomposing w_1u as $w_1u = v_1 \sqcup\!\sqcup_p v_2$ yields

$$v_1 = v_2 = a^{i_1}b^{j_1'}a^{n_1-i_1}b^{n_2-j_1'}.$$

Thus, as $|v_1|_a = n_1$ and $|v_2|_b = n_2$ we have $v_1 \in L_1$, $v_2 \in L_2$ and $w_1u \in L$.

Considering $w_2u = a^{2i_2}b^{2j'_2}a^{2(n_1-i_1)}b^{2(n_2-j'_1)}$, we can write $w_2u = v_3 \sqcup\!\sqcup_p v_4$ where

$$v_3 = v_4 = a^{i_2}b^{j'_2}a^{n_1-i_1}b^{n_2-j'_1}$$

and so $|v_3|_a = n_1 - i_1 + i_2$ and $|v_4|_b = n_2 - j'_1 + j'_2$. Our assumption that one of $i_1 \neq i_2$ and $j_1 \neq j_2$ is true implies that $v_3 = v_4 \notin L_1 \cap L_2$. Thus, $w_2u \notin L$.

Case (iii): j_1 even and j_2 odd. Let $0 \leq j'_1, j'_2 < n_2$ be integers such that $j_1 = 2j'_1$ and $j_2 = 2j'_2 + 1$.

Now $w_1u = a^{2i_1}b^{2j'_1}a^{2(n_1-i_1)}b^{2(n_2-j'_1)}$. As in case (ii), we have seen that $w_1u \in L$. Consider $w_2u = a^{2i_2}b^{2j'_2+1}a^{2(n_1-i_1)}b^{2(n_2-j'_1)}$. Thus $|w_2u| \equiv 1 \pmod 2$ and there do not exist words v_3, v_4 such that $v_3 \sqcup\!\sqcup_p v_4 = w_2u$.

Case (iv): j_1 odd and j_2 even. Let $0 \leq j'_1, j'_2 < n_2$ be integers such that $j_1 = 2j'_1 + 1$ and $j_2 = 2j'_2$. Consider $w_1u = a^{2i_1}b^{2j'_1+1}a^{2(n_1-i_1)}b^{2(n_2-j'_1)-1}$. Then as we have seen in case (i), $w_1u \in L$. However, consider $w_2u = a^{2i_1}b^{2j'_1}a^{2(n_1-i_1)}b^{2(n_2-j'_1)-1}$. As $|w_2u| \equiv 1 \pmod 2$, there do not exist words v_3, v_4 such that $w_2u = v_3 \sqcup\!\sqcup_p v_4$. ■

In the unary case, for any two words a^i, a^j, we have

$$a^i \sqcup\!\sqcup_p a^j = \begin{cases} a^{i+j} = a^{2i} & \text{if } i = j; \\ \emptyset & \text{otherwise.} \end{cases}$$

Thus, we see that for unary languages $L_1, L_2 \subseteq a^*$,

$$L_1 \sqcup\!\sqcup_p L_2 = h(L_1 \cap L_2)$$

where $h : a^* \to a^*$ is the morphism defined by $h(a) = a^2$. Thus, we can show that for unary languages

$$sc(L_1 \sqcup\!\sqcup_p L_2) = 2sc(L_1 \cap L_2).$$

The state complexity of intersection on unary languages is well-studied [155, 163, 202]. For instance, if $\gcd(n_1, n_2) = 1$, we can take $L_1 = (a^{n_1})^*$ and $L_2 = (a^{n_2})^*$ [184]. Thus, for these languages $sc(L_1 \sqcup\!\sqcup_p L_2) = 2n_1n_2$. However, if $\gcd(n_1, n_2) > 1$, the situation is more interesting.

For this case, see Pighizzini and Shallit [163]. We also note the work of Nicaud on the average state complexity of intersection [155].

4.3.2 Bounds on Slender Trajectories

We may now relate slenderness of sets of trajectories to state complexity. Our first result handles the case where $T = uv^*$.

In what follows, if u is a word of length n, then $u(i)$ represents the $(i+1)$-st letter of u for all $0 \le i \le n-1$. Further, let $\mathbf{n} = \{0, 1, 2, \ldots, n-1\}$.

Lemma 4.3.4 *Let $T = uv^*$ where $u, v \in \{0,1\}^*$. Let L_i be regular languages over Σ, with $sc(L_i) = n_i$, $i = 1, 2$. Let $L = L_1 \sqcup\!\sqcup_T L_2$. Then*

$$sc(L) \le |uv| n_1 n_2. \tag{4.1}$$

Proof. For $i = 1, 2$, let L_i be accepted by a DFA $M_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$ with $|Q_i| = n_i$. We describe $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = L_1 \sqcup\!\sqcup_T L_2$.

Let $n = |uv|$. We let $Q = Q_1 \times Q_2 \times \mathbf{n}$, $q_0 = [q_1, q_2, 0]$, and give δ by

$$\delta([q_i, q_j, k], a) = \begin{cases} [\delta_1(q_i, a), q_j, k+1] & \text{if} \quad (uv)(k) = 0 \quad \text{and} \quad k < n-1; \\ [\delta_1(q_i, a), q_j, |u|] & \text{if} \quad (uv)(k) = 0 \quad \text{and} \quad k = n-1; \\ [q_i, \delta_2(q_j, a), k+1] & \text{if} \quad (uv)(k) = 1 \quad \text{and} \quad k < n-1; \\ [q_i, \delta_2(q_j, a), |u|] & \text{if} \quad (uv)(k) = 1 \quad \text{and} \quad k = n-1. \end{cases}$$

Finally we let $F = F_1 \times F_2 \times \{|u|\}$. It is easily verified that $L(M)$ accepts the desired language. ■

We now give a bound for sets of trajectories $T = uv^*w$ with $w \neq \epsilon$.

Lemma 4.3.5 *Let $T = uv^*w$ where $u, v, w \in \{0,1\}^*$ and $w \neq \epsilon$. Let L_i be regular languages over Σ, with $sc(L_i) = n_i$, $i = 1, 2$. Let $L = L_1 \sqcup\!\sqcup_T L_2$. Then*

$$sc(L) \le n_1 n_2 \left(|u| + 1 + |v| \frac{(n_1 n_2)^{\left\lceil \frac{|w|}{|v|} \right\rceil + 1} - n_1 n_2}{n_1 n_2 - 1} \right). \tag{4.2}$$

Proof. For $i = 1, 2$, let L_i be accepted by a DFA $M_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$ with $|Q_i| = n_i$. We describe $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = L_1 \sqcup\!\sqcup_T L_2$.

Let $n = |u|$, $m = |v|$ and $s = |w|$. Let $b \notin \Sigma$ be a fixed new letter. We choose

$$Q = Q_1 \times Q_2 \times \{\mathbf{n} \cup \{b\}\} \cup Q_1 \times Q_2 \times \mathbf{m} \times \bigcup_{i=1}^{\lceil \frac{s}{m} \rceil} (Q_1 \times Q_2)^i. \tag{4.3}$$

Further, we let $q_0 = [q_1, q_2, 0] \in Q_1 \times Q_2 \times \mathbf{n}$.

For notational convenience, we define a set of functions $\gamma_{\alpha,\beta,a} : Q_1 \times Q_2 \to Q_1 \times Q_2$ for all $0 \leq \alpha \leq \lceil \frac{s}{m} \rceil - 1$, $0 \leq \beta < m$, $a \in \Sigma$, as follows

$$\gamma_{\alpha,\beta,a}([p_1, p_2]) = \begin{cases} [\delta_1(p_1, a), p_2)] & \text{if } w(m \cdot \alpha + \beta) = 0; \\ [p_1, \delta_2(p_2, a))] & \text{if } w(m \cdot \alpha + \beta) = 1; \end{cases}$$

for all $[p_1, p_2] \in Q_1 \times Q_2$. Further, we let $\gamma'_{\beta,a} : Q_1 \times Q_2 \to Q_1 \times Q_2$ be defined for all $0 \leq \beta < m$ and $a \in \Sigma$ by

$$\gamma'_{\beta,a}([q_i, q_j]) = \begin{cases} [\delta_1(q_i, a), q_j)] & \text{if } v(\beta) = 0; \\ [q_i, \delta_2(q_j, a))] & \text{if } v(\beta) = 1. \end{cases}$$

The full function δ is given by the following definitions. First, let $[q_i, q_j, k] \in Q_1 \times Q_2 \times \mathbf{n}$. Then,

$$\delta([q_i, q_j, k], a) = \begin{cases} [\delta_1(q_i, a), q_j, k+1] & \text{if} \quad k < n-1 \quad \text{and} \quad u(k) = 0; \\ [q_i, \delta_2(q_j, a), k+1] & \text{if} \quad k < n-1 \quad \text{and} \quad u(k) = 1; \\ [\delta_1(q_i, a), q_j, b] & \text{if} \quad k = n-1 \quad \text{and} \quad u(k) = 0; \\ [q_i, \delta_2(q_j, a), b] & \text{if} \quad k = n-1 \quad \text{and} \quad u(k) = 1. \end{cases} \tag{4.4}$$

If $[q_i, q_j, b] \in Q_1 \times Q_2 \times \{b\}$,

$$\delta([q_i, q_j, b], a) = [\gamma'_{0,a}(q_i, q_j), 1, \gamma_{0,0,a}(q_i, q_j)] \in Q_1 \times Q_2 \times \mathbf{m} \times Q_1 \times Q_2. \tag{4.5}$$

Now we can define δ on the set $Q_1 \times Q_2 \times \mathbf{m} \times \bigcup_{i=1}^{\lceil \frac{s}{m} \rceil} (Q_1 \times Q_2)^i$. Let $r \leq \lceil \frac{s}{m} \rceil$.

$$\delta([q_i, q_j, k, p_1^{(1)}, p_2^{(1)}, \ldots, p_1^{(r)}, p_2^{(r)}], a)$$
$$= \begin{cases} [\gamma'_{k,a}(q_i, q_j), \quad k+1, \gamma_{0,k,a}(p_1^{(1)}, p_2^{(1)}), \ldots, \gamma_{r-1,k,a}(p_1^{(r)}, p_2^{(r)})], \\ \qquad \text{if } 0 < k < m-1; \\ [\gamma'_{k,a}(q_i, q_j), \quad 0, \gamma_{0,k,a}(p_1^{(1)}, p_2^{(1)}), \ldots, \gamma_{r-1,k,a}(p_1^{(r)}, p_2^{(r)})], \\ \qquad \text{if } k = m-1; \\ [(\gamma'_{0,a}(q_i, q_j), \quad 1, \gamma_{0,k,a}(q_i, q_j), \gamma_{1,k,a}(p_1^{(1)}, p_2^{(1)}), \ldots, \gamma_{r,k,a}(p_1^{(r)}, p_2^{(r)})], \\ \qquad \text{if } k = 0, r < \lceil s/m \rceil; \\ [(\gamma'_{0,a}(q_i, q_j), \quad 1, \gamma_{0,k,a}(q_i, q_j), \gamma_{1,k,a}(p_1^{(1)}, p_2^{(1)}), \ldots, \gamma_{r-1,k,a}(p_1^{(r-1)}, p_2^{(r-1)})], \\ \qquad \text{if } k = 0, r = \lceil s/m \rceil. \end{cases}$$

The letter b distinguishes the case when we have not read any copies of v or w. We need a special letter to indicate this is the situation.

Let $f \in \mathbf{m}$ be chosen so that $f \equiv s \pmod{m}$. With this, we can define F by

$$F = Q_1 \times Q_2 \times f \times (Q_1 \times Q_2)^{\lceil s/m \rceil - 1} \times F_1 \times F_2.$$

Intuitively, we can explain the construction of M as follows. We note that the above parallel branches $[p_1^{(j)}, p_2^{(j)}]$, simulating a computation along w, are always separated by exactly m input letters. Thus in a state

$$[q_i, q_j, i, p_1^{(1)}, p_2^{(1)}, \ldots, p_1^{(r)}, p_2^{(r)}], \quad r \leq \lceil \frac{s}{m} \rceil, \tag{4.6}$$

the index i can keep track of the positions also of the r parallel branches along the suffix w of T: the ℓ-th pair is reading the $((\ell - 1) \cdot m + i)$-th letter of w.

When the index i goes from $m-1$ to zero, for each $1 \leq j \leq r-1$ the j-th pair of states $[p_1^{(j)}, p_2^{(j)}]$ is shifted into the $(j+1)$-st position (at the same time performing the appropriate state transition simulating M_1 or M_2). The first pair $[p_1^{(1)}, p_2^{(1)}]$ will then be added (based on the states $[q_i, q_j]$) to simulate the new computation that branches out from the loop v and into the suffix w.

The r-th computation is terminated when it reaches the end of w, that is, after s computation steps. Thus, we can have at most $\lceil s/m \rceil$ active computations on the suffix w of the trajectory.

Note that the transition function of M can implicitly code the word w as follows. When applying the transition function to a pair $[p_1^{(j)}, p_2^{(j)}]$, $1 \le j \le r$, and knowing the index i (in the notations of (4.6)), the indices i and j exactly specify the position in the word w. Thus M knows whether this position in w is a 0 or a 1 and can simulate a computation step of M_1 or M_2, respectively. This is implied by the definition of the functions $\gamma_{\alpha,\beta,a}$. ■

The following corollary follows easily by induction, noting that

$$L_1 ⧢_{T_1 \cup T_2} L_2 = (L_1 ⧢_{T_1} L_2) \cup (L_1 ⧢_{T_2} L_2).$$

Corollary 4.3.6 *Let $T \subseteq \{0, 1\}^*$ be a slender regular language with USL-index t, and write*

$$T = \bigcup_{i=1}^{t} u_i v_i^* w_i.$$

Then there exists a function K, depending only on the integers $|u_i|, |v_i|, |w_i|$, $1 \le i \le t$, such that

$$sc(L_1 ⧢_T L_2) \le K(sc(L_1)sc(L_2))^{t+s}$$

where

$$s = \sum_{i=1}^{t} \left\lceil \frac{|w_i|}{|v_i|} \right\rceil .$$

Our aim is to obtain a lower bound for the shuffle operation on trajectories with USL index 1. It seems likely that the bound (4.2) cannot be reached for any fixed set of trajectories (and for all values of $sc(L_i)$,$i = 1, 2$). In particular, if $|w|$ is fixed and $sc(L_i)$ can grow arbitrarily, then it seems impossible that the $\left\lceil \frac{|w|}{|v|} \right\rceil$ parallel computations on the suffix w could simultaneously reach all combinations of states of the DFAs for L_1 and L_2. Note that if the computation of M contains parallel branches that simulate the computations of M_i ($1 \le i \le 2$), in states $P_i \subseteq Q_i$, then all the states of P_i need to be reachable from a single state of M_i with inputs of length at most $|w|$.

For the above reason, we consider a lower bound for sets of trajectories uv^*w where the length of v and of w can depend on the sizes of the minimal DFAs for the component languages L_1 and

L_2. Furthermore, to simplify the notations below we give lower bound results for sets of trajectories of the form v^*w, i.e., $u = \epsilon$. It would be straightforward to modify the construction for prefixes u of arbitrary length to include the additive term $n_1n_2 \cdot (|u| + 1)$ from (4.2).

Lemma 4.3.7 *Let $\Sigma = \{a, b, c\}$. For any $n_1, n_2 \in \mathbb{N}$ there exist regular languages $L_i \subseteq \Sigma^*$ with $sc(L_i) = n_i$, $i = 1, 2$, and a set of trajectories $T = v^*w$, where $v, w \in \{0, 1\}^*$, such that*

$$sc(L_1 ⧢_T L_2) \geq (n_1n_2)^{\lceil \frac{|w|}{|v|} \rceil + 1}.$$

The ratio $|w|/|v|$ above can be chosen to be arbitrarily large.

Proof. Let L_1, L_2 be defined as $L_1 = \{w \in \Sigma^* : |w|_a \equiv 0 \pmod{n_1}\}$ and, $L_2 = \{w \in \Sigma^* : |w|_b \equiv 0 \pmod{n_2}\}$. Clearly $\text{sc}(L_i) = n_i$, $i = 1, 2$. Denote

$$n = \max(n_1, n_2) - 1 \text{ and } m = 2n.$$

For the set of trajectories we choose

$$T = ((01)^m)^*(10)^{mk}, \quad k \geq 1. \tag{4.7}$$

Note that $sc(T) = 2m(k + 1)$. Define $L = L_1 ⧢_T L_2$. The set $S \subseteq \Sigma^{(2k+1)m}$ is defined to consist of all words

$$S = \{w_1 \cdots w_{k+1} \ : \\ w_i \in \{a, c\}^m ⧢_p \{b, c\}^m, 1 \leq i \leq k, \ w_{k+1} \in \{a, c\}^n ⧢_p \{b, c\}^n\}. \tag{4.8}$$

If $w \in S$, then we denote by $w_1, w_2, \ldots, w_{k+1}$ the unique components of w as described by (4.8).

For $w \in S$ and $1 \leq i \leq k + 1$, we define the following quantities

$$A(w, a, i) = (\sum_{j=1}^{i} |w_j|_a) \bmod n_1, \quad A(w, b, i) = (\sum_{j=1}^{i} |w_j|_b) \bmod n_2.$$

Claim 4.3.8 *Let $w, w' \in S$. If there exists $1 \leq i \leq k + 1$ such that*

$$[A(w, a, i), A(w, b, i)] \neq [A(w', a, i), A(w', b, i)] \tag{4.9}$$

then $w \not\equiv_L w'$.

Proof. Assume i exists such that (4.9) holds and let $x_i \in \mathbf{n_i}$, $i = 1, 2$, be the integers such that

$$x_1 \equiv -A(w, a, i) \pmod{n_1} \text{ and } x_2 \equiv -A(w, b, i) \pmod{n_2}.$$

Choose

$$u_i = \begin{cases} ((b^{x_2}c^{n-x_2}) ⧢_p (a^{x_1}c^{n-x_1}))c^{2m(i-1)} & \text{if } i \leq k, \\ ((a^{x_1}c^{n-x_1}) ⧢_p (b^{x_2}c^{n-x_2}))c^{2mk} & \text{if } i = k+1. \end{cases}$$

To establish our claim it is sufficient to show that

$$wu_i \in L \text{ and } w'u_i \notin L. \tag{4.10}$$

Let $w = w_1 \cdots w_{k+1}$, $w' = w'_1 \cdots w'_{k+1} \in S$ be such that (4.9) holds for some index i. For each $1 \leq j \leq k$, let $w_j = \Omega_j ⧢_p \Pi_j$ and $w'_j = \Omega'_j ⧢_p \Pi'_j$ where $\Omega_j, \Omega'_j \in \{a, c\}^m$, $\Pi_j, \Pi'_j \in \{b, c\}^m$, and let $w_{k+1} = \Omega_{k+1} ⧢_p \Pi_{k+1}$ and $w'_{k+1} = \Omega'_{k+1} ⧢_p \Pi'_{k+1}$ where $\Omega_{k+1}, \Omega'_{k+1} \in \{a, c\}^n$, $\Pi_{k+1}, \Pi'_{k+1} \in \{b, c\}^n$.

(i) First we consider the case where $i \leq k$. Now $|wu_i| = |w'u_i| = 2m(k+i)$, so the only possible trajectory $t \in T$ which could correspond to these words is $t = (01)^{m \cdot i}(10)^{m \cdot k}$. Let $t = t_1t_2$ where $t_1 = (01)^{m \cdot i}$ and $t_2 = (10)^{m \cdot k}$. Let $\alpha, \alpha', \beta, \beta'$ be the unique words such that $\alpha ⧢_t \beta = wu_i$ and $\alpha' ⧢_t \beta' = w'u_i$. In particular, let $\alpha = \alpha_1\alpha_2$, $\alpha' = \alpha'_1\alpha'_2$, $\beta = \beta_1\beta_2$ and $\beta' = \beta'_1\beta'_2$ such that

$$\begin{aligned} w_1 \cdots w_i &= \alpha_1 ⧢_{t_1} \beta_1; \\ w'_1 \cdots w'_i &= \alpha'_1 ⧢_{t_1} \beta'_1; \\ w_{i+1} \cdots w_{k+1}u_i &= \alpha_2 ⧢_{t_2} \beta_2; \\ w'_{i+1} \cdots w'_{k+1}u_i &= \alpha'_2 ⧢_{t_2} \beta'_2. \end{aligned}$$

Then note that necessarily

$$\begin{aligned} \alpha_1 &= \Omega_1\Omega_2 \cdots \Omega_i; \\ \beta_1 &= \Pi_1\Pi_2 \cdots \Pi_i; \\ \alpha'_1 &= \Omega'_1\Omega'_2 \cdots \Omega'_i; \\ \beta'_1 &= \Pi'_1\Pi'_2 \cdots \Pi'_i. \end{aligned}$$

and

$$\begin{aligned}
\beta_2 &= \Omega_{i+1}\Omega_{i+2}\cdots\Omega_{k+1}b^{x_2}c^{n-x_2}c^{m(i-1)};\\
\alpha_2 &= \Pi_{i+1}\Pi_{i+2}\cdots\Pi_{k+1}a^{x_1}c^{n-x_1}c^{m(i-1)};\\
\beta_2' &= \Omega_{i+1}'\Omega_{i+2}'\cdots\Omega_{k+1}'b^{x_2}c^{n-x_2}c^{m(i-1)};\\
\alpha_2' &= \Pi_{i+1}'\Pi_{i+2}'\cdots\Pi_{k+1}'a^{x_1}c^{n-x_1}c^{m(i-1)}.
\end{aligned}$$

Thus, we can now easily compute $|\alpha|_a$, $|\alpha'|_a$, $|\beta|_b$, $|\beta'|_b$.

$$\begin{aligned}
|\alpha|_a &= |\alpha_1|_a + |\alpha_2|_a\\
&= A(w,a,i) + |\alpha_2|_a\\
&= A(w,a,i) + |\Pi_{i+1}\Pi_{i+2}\cdots\Pi_{k+1}|_a + x_1\\
&= A(w,a,i) + x_1 \equiv 0 \pmod{n_1}.
\end{aligned}$$

as $\Pi_j \in \{b,c\}^*$ and $x_1 \equiv -A(w,a,i) \pmod{n_1}$. An identical analysis yields that $|\alpha'|_a \equiv A(w',a,i) - A(w,a,i) \pmod{n_1}$. We can similarly examine β and β', to give

$$\begin{aligned}
|\beta|_b &\equiv 0 \pmod{n_2}\\
|\beta'|_b &\equiv A(w',b,i) - A(w,b,i) \pmod{n_2}
\end{aligned}$$

The congruences $|\alpha|_a \equiv 0 \pmod{n_1}$, $|\beta|_b \equiv 0 \pmod{n_2}$ give $wu_i \in L$. By (4.9), we conclude that one of $\alpha' \notin L_1$, $\beta' \notin L_2$ holds, and thus $w'u_i \notin L$.

(ii) Second we consider the case $i = k+1$. Now $|wu_i| = |w'u_i| = 2m(2k+1)$, so the corresponding trajectory is $t = t_1t_2$ where $t_1 = (01)^{m(k+1)}$ and $t_2 = (10)^{m\cdot k}$. In this case recall that $u_{k+1} = ((a^{x_1}c^{n-x_1}) \sqcup\!\sqcup_p (b^{x_2}c^{n-x_2}))c^{2mk}$, and the suffix c^{2mk} of u_{k+1} corresponds exactly to the suffix t_2 of the trajectory. Thus when the word wu_i (respectively, $w'u_i$) is written in the form $\alpha \sqcup\!\sqcup_t \beta$ (respectively, $\alpha' \sqcup\!\sqcup_t \beta'$) all letters a in the word correspond to ("come from") the component in L_1 and all letters b correspond to the component in L_2. By (4.9), we conclude that $\alpha \in L_1$ and $\beta \in L_2$ but necessarily one of $\alpha' \notin L_1$ or $\beta' \notin L_2$ holds. The completes the proof that (4.10) holds. ■

We now continue with the proof of Lemma 4.3.7. We claim that the map $\varphi : S \to (\mathbf{n_1} \times \mathbf{n_2})^{k+1}$, given by

$$w \mapsto [A(w, a, i), A(w, b, i)]_{i=1}^{k+1}, \tag{4.11}$$

is surjective. To see this, note that if $w \in S$ then $A(w, a, i)$ and $A(w, b, i)$ depend only on the subwords $w_1, \ldots, w_i$. Thus after $w_1, \ldots, w_i$ are chosen we can always select an arbitrary value for $[A(w, a, i+1), A(w, b, i+1)]$ since $[|w_{i+1}|_a, |w_{i+1}|_b]$ can have any value in $\mathbf{n_1} \times \mathbf{n_2}$. (This holds also in the case $i = k$.) Thus, φ is surjective, and by Claim 4.3.8, for distinct $z, z' \in (\mathbf{n_1} \times \mathbf{n_2})^{k+1}$, the sets $\varphi^{-1}(z)$ and $\varphi^{-1}(z')$ lie in different equivalence classes of $\equiv_L$. Thus, $sc(L) \geq (n_1 n_2)^{k+1}$. ■

In the notations of Lemma 4.3.7, the upper bound (4.2) is of the order $|v| \cdot (n_1 n_2)^{\lceil \frac{|w|}{|v|} \rceil + 1}$ where $|v|$ can be chosen as a constant times $\max(n_1, n_2)$. In the proof of Lemma 4.3.7 we counted only equivalence classes of $\equiv_L$ that had representatives of length $(2k+1)m$. Using the same construction we can get an improved lower bound by taking into account also equivalence classes with representatives of different lengths. This bound approaches the upper bound when $|v|$ grows compared to $sc(L_i), i = 1, 2$.

Lemma 4.3.9 *Let $\Sigma = \{a, b, c\}$. Let $n_1, n_2 \in \mathbb{N}$ be arbitrary and $n = \max(n_1, n_2) - 1$. There exist regular languages $L_i \subseteq \Sigma^*$ with $sc(L_i) = n_i$, $i = 1, 2$, and a set of trajectories $T = v^* w$, where $v, w \in \{0, 1\}^*$, $|v| \geq 4n$, such that*

$$sc(L_1 \sqcup\!\sqcup_T L_2) \geq (|v| - 4n + 1)(n_1 n_2)^{\lceil \frac{|w|}{|v|} \rceil + 1} + |v| \frac{(n_1 n_2)^{\lceil \frac{|w|}{|v|} \rceil - 1} - 1}{n_1 n_2 - 1}. \tag{4.12}$$

The quantity $|v|$ and the ratio $|w|/|v|$ above can be chosen to be arbitrarily large compared to $sc(L_1)$ and $sc(L_2)$.

Proof. We use the notations from the proof of Lemma 4.3.7 with the only change that $m \geq 2n$ can be arbitrary (instead of $m = 2n$).

For a word w with $|w| \leq 2m(k+1)$ and $w = w_1 \cdots w_{i-1} w_i$ where $|w_j| = 2m$, $j = 1, \ldots, i-1$, $0 \leq |w_i| \leq 2m$, we say that the jth component of w is w_j, $j = 1, \ldots, i$.

Since T consists of only words with length a multiple of $2m$, it follows that for any $w, w' \in \Sigma^*$ if $|w| \not\equiv |w'| \pmod{2m}$ then $w \not\equiv_L w'$. Note that any word in Σ^* can be completed to a word in L by adding a suitable suffix. On the other hand, if $w, w' \in S$ and $w \not\equiv_L w'$ then

$$wc^j \not\equiv_L w'c^j \text{ for all } 1 \leq j \leq 2m - 4n. \tag{4.13}$$

Note that $\varphi(w) \neq \varphi(w')$ and the suffix c^j does not change the numbers of occurrences of a's and b's in the $(k+1)$-st component. Furthermore, we can always find a word u of length $2m - 2n - j \, (\geq 2n)$ such that $wc^j u \in L$ (or $w'c^j u \in L$) which may be needed to establish the inequivalence of wc^j and $w'c^j$ if $\varphi(w)$ and $\varphi(w')$ differ only in their first component.

Since we know that S contains $(n_1 n_2)^{\lceil \frac{|w|}{|v|} \rceil + 1}$ pairwise inequivalent words, the above observations give us $(2m - 4n + 1)(n_1 n_2)^{\lceil \frac{|w|}{|v|} \rceil + 1}$ equivalence classes which is the first term of (4.12).

Let S_i, $1 \leq i \leq k$, denote the set of prefixes of S having length $2mi$. Similarly as in the proof of Lemma 4.3.7, we see that S_i contains representatives of $(n_1 n_2)^i$ distinct equivalence classes of $\equiv_L$. Using a similar argument as above for (4.13) we see that if $w, w' \in S_i$, $i < k$, and $w \not\equiv_L w'$ then $wc^j \not\equiv_L w'c^j$ for all $1 \leq j < 2m$. Note that since $i < k$ the suffix c^j does not belong to the $(k+1)$-st component and it can have any length up to $2m$. Furthermore, each word

$$wc^j, \quad w \in S_i, \; i \leq k - 2, \; 0 \leq j < 2m \tag{4.14}$$

can be completed to a word in L using a suffix of length $2m(k - i) - j$ and not by any suffix of shorter length. Thus any two words of different length as in (4.14) cannot be equivalent, and any word as in (4.14) cannot be equivalent to any word as in (4.13). This yields

$$2m \sum_{i=0}^{k-2} (n_1 n_2)^i$$

equivalence classes which is the last term of (4.12). ■

As a consequence of Lemma 4.3.9 we have:

Theorem 4.3.10 *The upper bound (4.2) is asymptotically optimal if $sc(T)$ (that is, $|v|$) can be arbitrarily large compared to $sc(L_i)$, $i = 1, 2$.*

In comparing Theorem 4.3.10 and Lemma 4.3.3, we note that Lemma 4.3.3 is a tighter bound, and is better than Theorem 4.3.10 in the restricted sense that Lemma 4.3.3 takes a set of trajectories (albeit a very specific and fixed set of trajectories) and defines languages for which we have matching upper and lower bounds. This is subtly different from Theorem 4.3.10, which takes languages and defines a set of trajectories for which the upper bound is obtained. The reasoning for this, we recall, is discussed prior to the statement of Lemma 4.3.7.

4.4 Future Directions

4.4.1 Polynomial Density Trajectories

We may also consider the case of polynomial-density sets of trajectories, i.e., sets of trajectories T with $p_T(n) \in O(n^k)$ for $k \geq 1$, by extending the ideas of Lemma 4.3.5. We can employ nondeterministic state complexity when it is to our advantage. However, the upper bound which we obtain is not much better than the bound of Lemma 4.2.2. We note that an extension to linear density sets of trajectories would encompass the case of $T = 0^*1^*$. By Theorem 4.2.4, we know that this linear density bound would not be as good an improvement over Lemma 4.2.2 compared to, e.g., Corollary 4.3.6.

4.4.2 Exponential Density Trajectories

Recall that the example of arbitrary shuffle, shown by Câmpeanu *et al.* to have state complexity no better than our construction in Lemma 4.2.2, uses the set of trajectories $T = (0 + 1)^*$ of density 2^n. We also note that, by Szilard *et al.* [190], the density of a regular language over Σ is either $O(p(n))$, where p is a polynomial, or $\Omega(|\Sigma|^n)$.

Thus, we may conjecture that a set of trajectories T yields an operation which is, in the worst case, no better than Lemma 4.2.2 only in the case when $p_T(n) \in \Omega(2^n)$, i.e. T has exponential density.

4.4.3 Other Open Problems

Our constructions in Lemmas 4.3.7 and 4.3.9 use three-letter alphabets. Can these constructions be improved to two-letter alphabets? The problem of restricting the alphabet size to be as small as possible is often challenging. For example, in the case of concatenation, the state complexity problem was solved for a three-letter alphabet by Yu *et al.* [204], but the case of a two-letter alphabet was open until very recently [95, 94].

4.5 Conclusions

In this chapter we have examined the state complexity of shuffle on trajectories. This area has been previously examined by Câmpeanu *et al.* [22] for the case of $T = \{0, 1\}^*$, and by Yu *et al.* [204] for the case of $T = 0^*1^*$. In this chapter, we have considered state complexity of arbitrary shuffle on trajectories.

We have also considered the specific case where the set T of trajectories is slender, i.e., contains only a constant number of words of each length. In this case, we have shown that shuffle on the set T of trajectories has a considerably lower state complexity than in the case of a general $T \subseteq \{0, 1\}^*$.