State Complexity of Simple Splicing

Lila Kari and Timothy Ng

School of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada
{lila.kari, tim.ng}@uwaterloo.ca

Abstract. Splicing, as a binary word/language operation, was inspired by the DNA recombination under the action of restriction enzymes and ligases, and was first introduced by Tom Head in 1987. Splicing systems as generative mechanisms were defined as consisting of an initial starting set of words called an axiom set, and a set of splicing rules—each encoding a splicing operation—, as their computational engine to iteratively generate new strings starting from the axiom set. Since finite splicing systems (splicing systems with a finite axiom set and a finite set of splicing rules) generate a subclass of the family of regular languages, descriptional complexity questions about splicing systems can be answered in terms of the size of the minimal deterministic finite automata that recognize their languages. In this paper we focus on a particular type of splicing systems, called simple splicing systems, where the splicing rules are of a particular form. We prove a tight state complexity bound of $2^n - 1$ for (semi-)simple splicing systems with a regular initial language with state complexity $n \geq 3$. We also show that the state complexity of a (semi-)simple splicing system with a finite initial language is at most $2^{n-2} + 1$, and that whether this bound is reachable or not depends on the size of the alphabet and the number of splicing rules.

1 Introduction

In [10] Head described a language-theoretic operation, called *splicing*, which models DNA recombination, a cut-and-paste operation on DNA double-stranded molecules, under the action of restriction enzymes and ligases. A *splicing system* is a formal language model which consists of a set of *initial words*, I (representing double-stranded DNA strings), and a set of *splicing rules* R (representing restriction enzymes). The most commonly used definition for a splicing rule is a quadruplet of words $r = (u_1, v_1; u_2, v_2)$. This rule splices two words $x_1u_1v_1y_1$ and $x_2u_2v_2y_2$: the words are cut between the factors u_1, v_1, respectively u_2, v_2, and the prefix (the left segment) of the first word is recombined by catenation with the suffix (the right segment) of the second word; see Figure 1 and also [16]. The words u_1v_1 and u_2v_2 are the restriction sites in the rule r. A splicing system generates a language which contains every word that can be obtained by successively applying rules to axioms and the intermediately produced words. The most natural variant of splicing systems, often referred to as *finite splicing systems*, is to consider a finite set of axioms and a finite set of rules.

Several different types of splicing systems have been proposed in the literature, and Bonizzoni et al. [1] showed that the classes of languages they generate are related. Shortly after the introduction of splicing in formal language theory, Culik II and Harju [4] proved that finite splicing systems can only generate regular languages; see also [11, 15]. Gatterdam [7] gave $(aa)^*$ as an example of a regular language which cannot be generated by a finite splicing system; thus, the class of languages generated by finite splicing systems is strictly included in the class of regular languages.

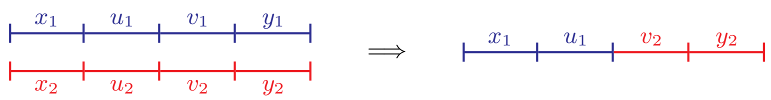

Fig. 1. Splicing of the words $x_1u_1v_1y_1$ and $x_2u_2v_2y_2$ by the rule $r = (u_1, v_1; u_2, v_2)$.

Descriptional complexity considers the complexity of a language in terms of the size of a computational device (in this case splicing system) that generates or recognizes it. For instance, Mateescu et al. [14] consider a number of descriptional complexity measures for simple splicing systems, such as the number of rules, the number of words in the initial language, the maximum length of a word in the initial language, and the sum of the lengths of all words in the initial language. Loos et al. [13] consider the descriptional complexity of finite splicing systems by using the number of rules, the length of the rules, and the size of the initial language as complexity measures. Păun [16] proposed using the radius, the largest u_i in a rule, as a descriptional complexity measure.

As the class of languages generated by splicing systems forms a subclass of the family of regular languages, their descriptional complexity can also be considered in terms of the finite automata that recognize them. For example, Loos et al. [13] gave a bound on the number of states required for a nondeterministic finite automaton to recognize the language generated by an equivalent finite splicing system.

We focus our attention on simple splicing systems, that is, splicing systems where the rules $(u_1, v_1; u_2, v_2)$ are of a particular form: $u_1 = u_2 = a$, are singleton letters, and $v_1 = v_2 = \varepsilon$ are the empty word. The descriptional complexity of simple splicing systems was considered by Mateescu et al. [14] in terms of the size of a right linear grammar that generates a simple splicing language. Here we consider the descriptional complexity of simple splicing systems in terms of deterministic state complexity [6]. In other words, we measure the descriptional complexity of a simple splicing system in terms of the size of the minimal deterministic finite automaton that recognizes the language generated by the splicing system.

In this paper, we prove tight state complexity bounds for simple and semi-simple splicing systems with regular and finite initial languages. In Section 2, we fix notation and definitions for simple splicing systems. We consider the state complexity of simple splicing systems with regular and finite initial languages in Section 3. In Section 4, we give tight state complexity bounds for semi-simple splicing systems with finite initial languages. We consider the state complexity of the crossover operation related to simple splicing systems in Section 5.

2 Preliminaries

Let Σ be a finite alphabet. We denote by Σ^* the set of all finite words over Σ, including the empty word, which we denote by ε. We denote the length of a word w by $|w| = n$. If $w = xyz$ for $x, y, z \in \Sigma^*$, we say that x is a prefix of w, y is a factor of w, and z is a suffix of w.

A deterministic finite automaton (DFA) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, δ is a function $\delta : Q \times \Sigma \to Q$, $s \in Q$ is the initial state, and $F \subset Q$ is a set of final states. We extend the transition function δ to a function $Q \times \Sigma^* \to Q$ in the usual way. A DFA A is complete if δ is defined for all $q \in Q$ and $a \in \Sigma$. We will make use of the notation $q \xrightarrow{w} q'$ for $\delta(q, w) = q'$, where $w \in \Sigma^*$ and $q, q' \in Q$. A state $q \in Q$ is called a sink state if $\delta(q, a) = q$ for all $a \in \Sigma$ and $q \notin F$.

Each letter $a \in \Sigma$ defines a transformation of the state set Q. Let $\delta_a : Q \to Q$ be the transformation on Q induced by a, defined by $\delta_a(q) = \delta(q, a)$. We extend this definition to words by composing the transformations $\delta_w = \delta_{a_1} \circ \delta_{a_2} \circ \cdots \circ \delta_{a_n}$ for $w = a_1 a_2 \cdots a_n$. We denote by $\operatorname{im} \delta_a$ the image of δ_a, defined $\operatorname{im} \delta_a = \{\delta(p, a) \mid p \in Q\}$.

The language recognized or accepted by A is $L(A) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. A state q is called *reachable* if there exists a string $w \in \Sigma^*$ such that $\delta(q_0, w) = q$. Two states p and q of A are said to be *equivalent* if $\delta(p, w) \in F$ if and only if $\delta(q, w) \in F$ for every word $w \in \Sigma^*$. A DFA A is minimal if each state $q \in Q$ is reachable from the initial state and no two states are equivalent. The state complexity of a regular language L is the number of states of the minimal complete DFA recognizing L [6].

A nondeterministic finite automaton (NFA) is a tuple $A = (Q, \Sigma, \delta, I, F)$ where Q is a finite set of states, Σ is an alphabet, δ is a function $\delta : Q \times \Sigma \to 2^Q$, $I \subseteq Q$ is a set of initial states, and F is a set of final states. The language recognized by an NFA A is $L(A) = \{w \in \Sigma^* \mid \bigcup_{q \in I} \delta(q, w) \cap F \neq \emptyset\}$. As with DFAs, transitions of A can be viewed as transformations on the state set. Let $\delta_a : Q \to 2^Q$ be the transformation on Q induced by a, defined by $\delta_a(q) = \delta(q, a)$. The image of δ_a is defined by $\operatorname{im} \delta_a = \{\delta(p, a) \mid p \in Q\}$. We make use of the notation $P \xrightarrow{w} P'$ for $P' = \bigcup_{q \in P} \delta(q, w)$, where $w \in \Sigma^*$ and $P, P' \subseteq Q$.

2.1 Simple Splicing Systems

In this paper we will use the notation of Păun [16], even though simple splicing systems can be defined using any of the three definitions of splicing. The splicing operation is defined via sets of quadruples $r = (\alpha_1, \alpha_2; \alpha_3, \alpha_4)$ with $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \Sigma^*$ called splicing rules. For two strings $x = x_1 \alpha_1 \alpha_2 x_2$ and $y = y_1 \alpha_3 \alpha_4 y_2$, applying the rule $r = (\alpha_1, \alpha_2; \alpha_3, \alpha_4)$ produces a string $z = x_1\alpha_1\alpha_4 y_2$, which we denote by $(x, y) \vdash^r z$.

A *splicing scheme* is a pair $\sigma = (\Sigma, \mathcal{R})$ where Σ is an alphabet and $\mathcal{R}$ is a set of splicing rules. For a splicing scheme $\sigma = (\Sigma, \mathcal{R})$ and a language $L \subseteq \Sigma^*$, we denote by $\sigma(L)$ the language

$$\sigma(L) = L \cup \{z \in \Sigma^* \mid (x, y) \vdash^r z, \text{ where } x, y \in L, r \in \mathcal{R}\}.$$

Then we define $\sigma^0(L) = L$ and $\sigma^{i+1}(L) = \sigma(\sigma^i(L))$ for $i \geq 0$ and

$$\sigma^*(L) = \lim_{i\to\infty} \sigma^i(L) = \bigcup_{i\geq 0} \sigma^i(L).$$

For a splicing scheme $\sigma = (\Sigma, \mathcal{R})$ and an initial language $L \subseteq \Sigma^*$, we say the triple $H = (\Sigma, \mathcal{R}, L)$ is a *splicing system.* The language generated by H is defined by $L(H) = \sigma^*(L)$.

Mateescu et al. [14] define a restricted class of splicing systems called simple splicing systems. A *simple splicing system* is a triple $H = (\Sigma, M, I)$, where Σ is an alphabet, $M \subseteq \Sigma$ is a set of markers, and I is a finite initial language over Σ. For $a \in M$, we have $(x, y) \vdash^a z$ if and only if $x = x_1 a x_2$, $y = y_1 a y_2$, and $z = x_1 a y_2$ for some $x_1, x_2, y_1, y_2 \in \Sigma^*$.

In other words, a simple splicing system is a system in which the set of rules is $\mathcal{M} = \{(a, \varepsilon; a, \varepsilon) \mid a \in M\}$ and the initial language I is finite. Since the rules are determined solely by our choice of $M \subseteq \Sigma$, the set M is used in the definition of the simple splicing system rather than the set of rules $\mathcal{M}$. Based on these properties, one can deduce that the class of languages generated by simple splicing systems is subregular [4, 15]. Mateescu et al. [14] show that these languages form a proper subclass of the extended star-free languages.

In this paper, we will relax the condition that the initial language of a simple splicing system must be a finite language. We will consider also simple splicing systems with regular initial languages. By [16], it is clear that such a splicing system will also produce a regular language. In the following, we will use the convention that I denotes a finite language and L denotes an infinite language.

3 State Complexity of Simple Splicing

In this section, we will give tight state complexity bounds for simple splicing systems with regular and finite initial languages. First, we will define an NFA that recognizes the language of a given simple splicing system. The construction follows a more general construction due to Loos et al. [13] for finite splicing systems. This construction is a simplification of a construction by Pixton [15], which itself is a simplification of the original proof of regularity of finite splicing due to Culik and Harju [4].

Proposition 1. *Let $H = (\Sigma, M, L)$ be a simple splicing system with a regular initial language L and let L be recognized by a DFA with n states. Then there exists an NFA A'_H with n states such that $L(A'_H) = L(H)$.*

Proof. Let $H = (\Sigma, M, L)$ and let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. We will define the NFA $A_H = (Q', \Sigma, \delta', q_0, F)$, where $Q' = Q \cup Q_M$ with $Q_M = \{p_a, p'_a \mid a \in M\}$ and the transition function δ' is defined

- $\delta'(q, a) = \{\delta(q, a)\}$ if $q \in Q$ and $a \in \Sigma$,
- $\delta'(q, \varepsilon) = \{p_a\}$ if $q \in Q$, $a \in M$, and $\delta(q, a)$ is not the sink state,
- $\delta'(p_a, a) = \{p'_a\}$ if $p_a \in Q_M$,
- $\delta'(p'_a, \varepsilon) = \operatorname{im} \delta_a$ if $p'_a \in Q_M$ and $a \in M$

First, we describe the construction of [13]. Let $\mathcal{M} = \{(a, \varepsilon; a, \varepsilon) \mid a \in M\}$ be the set of rules for H. For each rule $(\alpha_1, \alpha_2; \alpha_3, \alpha_4) \in \mathcal{M}$, add new states and transitions corresponding to $\alpha_1\alpha_4$ and $\alpha_3\alpha_2$. That is, if $\alpha_1 = a_1 \cdots a_i$, $\alpha_2 = b_1 \cdots b_j$, $\alpha_3 = c_1 \cdots c_k$, and $\alpha_4 = d_1 \cdots d_\ell$, then add states and transitions corresponding to a path $r_0 \xrightarrow{a_1} r_1 \xrightarrow{a_2} \cdots \xrightarrow{d_\ell} r_{i+\ell}$ for $\alpha_1\alpha_4$ and a path $s_0 \xrightarrow{c_1} s_1 \xrightarrow{c_2} \cdots \xrightarrow{b_j} s_{j+k}$ corresponding to $\alpha_3\alpha_2$. Now consider each path $q \xrightarrow{\alpha_1\alpha_2} q'$ in A such that q is reachable from the initial state q_0 and a final state of A is reachable from q'. We add an ε-transition from q to r_0 and from s_{j+k} to q'. Similarly, for each path $t \xrightarrow{\alpha_3\alpha_4} t'$, add ε-transitions from t to s_0 and from $r_{i+\ell}$ to t'.

Now, since H is a simple splicing system, this construction can be simplified further. Since every rule of H is of the form $(a, \varepsilon; a, \varepsilon)$, we only need to add states and transitions for $p_a \xrightarrow{a} p'_a$ for each rule. Then add ε-transitions from states q of A to p_a if q has an outgoing transition on a to a non-sink state of A. From each state p'_a, add ε-transitions to each state of A with an incoming transition on a. Recall that $\operatorname{im} \delta_a$ is the image of the transformation of δ induced by a, and therefore it is the set of states of A with an incoming transition on a.

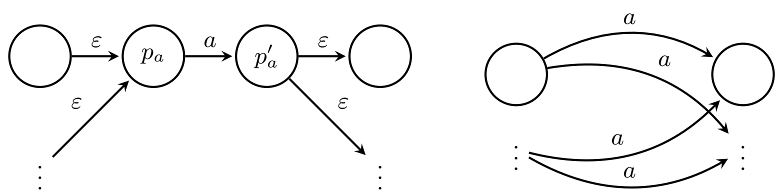

Fig. 2. New states and transitions for $a \in M$ (left), after ε-removal (right)

Finally, we can simplify this NFA by removing ε-transitions in the usual way to obtain an NFA $A'_H = (Q, \Sigma, \delta', q_0, F)$, where

$$\delta'(q, a) = \begin{cases} \{\delta(q, a)\} & \text{if } \delta(q, a) \text{ is the sink state,} \\ \{\delta(q, a)\} & \text{if } a \notin M, \\ \operatorname{im} \delta_a & \text{if } a \in M. \end{cases}$$

Figure 2 illustrates the new states and transitions added for $a \in M$ before and after ε-removal. Observe that by removing the ε-transitions, we also remove the states that were initially added earlier in the construction of A_H. Thus, the state set of A'_H is exactly the state set of the DFA A recognizing L. □

Given a splicing system $H = (\Sigma, M, L)$, one can obtain a DFA that recognizes $L(H)$ by performing the subset construction on A'_H. As shown in Proposition 1, if L is recognized by a DFA with n states, then A'_H also has n states. By applying the subset construction and observing that the empty set is not reachable from any subset of Q in A'_H, this gives an upper bound of $2^n - 1$ states for a DFA equivalent to A'_H.

We will now show that there exists a family of regular languages L_n with state complexity n such that a simple splicing system $H = (\Sigma, M, L_n)$ with one marker requires $2^n - 1$ states for an equivalent DFA to recognize it.

Proposition 2. *For $|\Sigma| \geq 3$ and $n \geq 3$, there exists a simple splicing system with a regular initial language $H = (\Sigma, M, L_n)$ with $|M| = 1$ where L_n is a regular language with state complexity n such that the minimal DFA for $L(H)$ requires at least $2^n - 1$ states.*

Proposition 2 is proved via the family of languages L_n accepted by DFAs A_n, shown in Figure 3, with $M = \{c\}$.

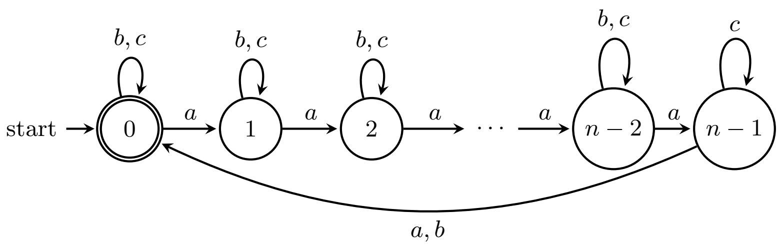

Fig. 3. The DFA A_n

Together, Propositions 1 and 2 give the following result.

Theorem 3. *For a simple splicing system with a regular initial language $H = (\Sigma, M, L_n)$ where $M \subseteq \Sigma$ and $L_n \subseteq \Sigma^*$ has state complexity n, the state complexity of $L(H)$ is at most $2^n - 1$ and this bound can be reached in the worst case.*

We will now consider simple splicing systems with a finite initial language. We will show that the upper bound of Proposition 1 is not reachable in this case.

Proposition 4. *Let $H = (\Sigma, M, I)$ be a simple splicing system with a finite initial language, where I is a finite language recognized by a DFA A with n states. Then a DFA recognizing $L(H)$ requires at most $2^{n-2} + 1$ states.*

We will show that this bound is reachable. We note that the lower bound witness used in the following lemma is defined over an alphabet with size exponential in the number of states of the DFA recognizing the initial language.

Lemma 5. *There exists a simple splicing system with a finite initial language $H = (\Sigma, M, I_n)$ where I_n is a finite language with state complexity n such that a DFA recognizing $L(H)$ requires $2^{n-2} + 1$ states.*

Together, Proposition 4 and Lemma 5 give the following result.

Theorem 6. *For a simple splicing system with a finite initial language $H = (\Sigma, M, I_n)$ where $M \subseteq \Sigma$ and $I_n \subseteq \Sigma^*$ has state complexity n, the state complexity of $L(H)$ is at most $2^{n-2} + 1$ and this bound can be reached in the worst case.*

The bound of Lemma 5 is reached by a witness defined over an alphabet size of $2^{n-3} + 1$. An obvious question is whether this bound can be reached via a smaller alphabet. We will consider in the following the state complexity of simple splicing systems with a finite initial language for small, fixed alphabets. We begin with a general observation on the transition function of a DFA recognizing the language of a simple splicing system.

Lemma 7. *Let $H = (\Sigma, M, L)$ be a simple splicing system with a regular initial language and let A_H be an NFA recognizing $L(H)$. If $a \in M$ and δ' is the transition function of A_H, then $|\operatorname{im} \delta'_a| = 2$.*

First, we will consider simple splicing systems with a finite initial language defined over a unary alphabet.

Proposition 8. *Let $H = (\{a\}, M, I)$ be a simple splicing system where M is nonempty and I is a finite language containing a word of length at least 2. Then the minimal DFA recognizing $L(H)$ has exactly two states.*

Next, we consider simple splicing systems with a finite initial language defined over a binary alphabet. We will show that the small size of the alphabet restricts the number of transformations that can be performed on the state set and that the upper bound on the number of states falls far below the upper bound of Proposition 4 as a result.

Proposition 9. *Let $H = (\{a, b\}, M, I)$ be a simple splicing system where I is a finite language with state complexity n. Then the minimal DFA recognizing $L(H)$ has at most $2n - 3$ states and this bound is reachable in the worst case.*

We will now consider the state complexity of simple splicing systems with a finite initial language defined over a ternary alphabet. We will show that the upper bound of $2^{n-2} + 1$ from Proposition 4 cannot be reached with an alphabet of size 3.

Proposition 10. *Let $H = (\{a, b, c\}, M, I)$ be a simple splicing system where I is a finite language with state complexity n. Then the minimal DFA recognizing $L(H)$ has at most $2^{\frac{n}{2}} + 1$ states if n is even and $3 \cdot 2^{\frac{n-3}{2}} + 1$ states if n is odd.*

We note that the upper bound of the previous lemma is similar to the state complexity of the reversal operation on finite languages [2]. We will use this result as inspiration for a family of lower bound witnesses in the following lemma.

Lemma 11. *There exists a family of finite languages $I_n \subseteq \{a,b,c\}^*$, for $n \geq 4$, recognized by a DFA with n states such that the minimal DFA for a simple splicing system $H = (\{a,b,c\}, M, I_n)$ requires at least $2^{\frac{n}{2}} + 1$ states if n is even and $3 \cdot 2^{\frac{n-3}{2}} + 1$ states if n is odd.*

The family of witness languages I_n used to prove Lemma 11 is accepted by DFAs A_n, shown in Figure 4, with $M = \{c\}$.

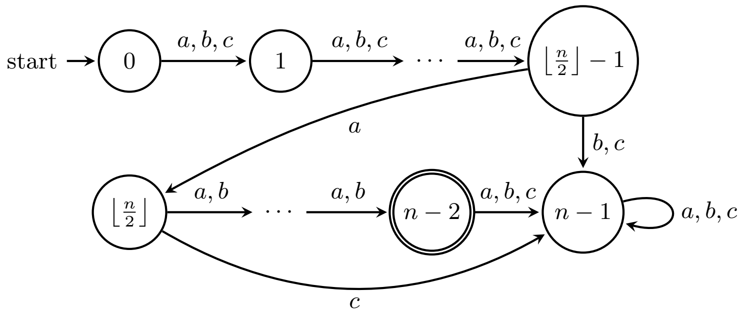

Fig. 4. The ternary witness DFA A_n

Together, Proposition 10 and Lemma 11 give us the following theorem.

Theorem 12. *For a simple splicing system with a finite initial language $H = (\Sigma, M, I_n)$ where $|\Sigma| = 3$, $M \subseteq \Sigma$, and $I_n \subseteq \Sigma^*$ has state complexity n, the state complexity of $L(H)$ is at most $2^{\frac{n}{2}} + 1$ states if n is even and $3 \cdot 2^{\frac{n-3}{2}} + 1$ states if n is odd and this bound can be reached in the worst case.*

4 State Complexity of Semi-simple Splicing

In this section, we will give tight state complexity bounds for semi-simple splicing systems with regular and finite initial languages. In particular, we will show that the upper bound is reachable for semi-simple splicing systems with a finite initial language defined over a fixed alphabet.

Goode and Pixton [9] generalize simple splicing systems by defining semi-simple splicing systems. A splicing system is semi-simple if every rule is of the form $(a, \varepsilon; b, \varepsilon)$ for $a, b \in \Sigma$. Again, rather than explicitly define a set of rules $\mathcal{M}$, we refer instead to the set $M^{(2)} \subseteq \Sigma \times \Sigma$ of pairs of symbols, which determines the set of rules. As with simple splicing systems, one can conclude that the class of languages generated by semi-simple splicing systems is subregular [4, 15].

In the following, we will give a construction for an NFA that recognizes the language generated by a semi-simple splicing system. As with the NFA for simple splicing systems from Proposition 1, the construction will follow the more general construction for finite splicing systems of Loos et al. [13].

Proposition 13. *Let $H = (\Sigma, M^{(2)}, L)$ be a semi-simple splicing system with a regular initial language. Then there exists an NFA B'_H with n states such that $L(B'_H) = L(H)$.*

It is clear from Proposition 13 that for a given regular language L, the language of a semi-simple splicing system $H = (\Sigma, M^{(2)}, L)$ can require $2^n - 1$ states in the worst case. Since a simple splicing system is also a semi-simple splicing system, the lower bound witness from Proposition 2 holds. Therefore, we can focus on the more interesting case of semi-simple splicing systems with finite initial languages. First, we observe that even with semi-simple splicing rules, the upper bound on the number of states for a DFA recognizing a semi-simple splicing system with a finite initial language remains the same.

Proposition 14. *Let $H = (\Sigma, M^{(2)}, I)$ be a semi-simple splicing system with a finite initial language where I is a finite language recognized by a DFA A with n states. Then a DFA recognizing $L(H)$ requires at most $2^{n-2} + 1$ states.*

The proof of this fact is identical to the proof of Proposition 4.

Recall from Lemma 5, that the lower bound witness for simple splicing systems with a finite initial language was defined over an alphabet with size exponential in the state complexity of the initial language. We will show in the following lemma that for semi-simple splicing systems with a finite initial language, a lower bound witness defined over an alphabet of size 3 exists.

Lemma 15. *Let $n \geq 4$. Then there exists a semi-simple splicing system with a finite initial language $H = (\Sigma, M^{(2)}, I_n)$ where $|\Sigma| = 3$ and I_n is a finite language with state complexity n such that $L(H)$ is recognized by a DFA that requires at least $2^{n-2} + 1$ states.*

The family of witness languages I_n of Lemma 15 is accepted by DFAs A_n, shown in Figure 5, with $\Sigma = \{a, b, c\}$ and $M^{(2)} = \{(a, c)\}$.

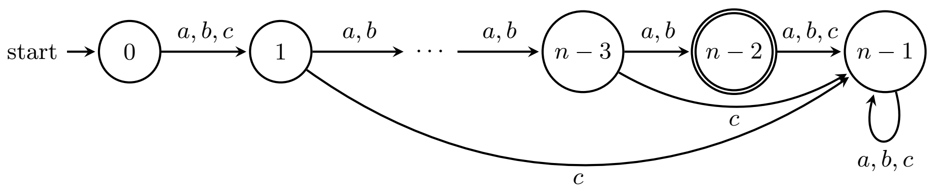

Fig. 5. The ternary witness DFA A_n

From Proposition 14 and Lemma 15, we have the following result.

Theorem 16. *For a semi-simple splicing system with a finite initial language $H = (\Sigma, M^{(2)}, I_n)$ where $M \subseteq \Sigma$ and $I_n \subseteq \Sigma^*$ has state complexity n, the state complexity of $L(H)$ is at most $2^{n-2} + 1$ and this bound can be reached in the worst case.*

5 State Complexity of the Crossover Operation

In this section, we will give tight state complexity bounds for the crossover operation [3], which can be thought of as a single step of semi-simple splicing. Mateescu et al. [14] gave an algebraic characterization of the class of languages generated by simple splicing systems based on the crossover operation therein. A similar such characterization for the class of languages generated by semi-simple splicing systems is given by Ceterchi [3].

For $M = M_1 \times M_2 \subseteq \Sigma \times \Sigma$, define the operation $\diamond_M$ on two strings $u, v \in \Sigma^+$ by

$$u \diamond_M v = \begin{cases} u'av' & \text{if } u = u'a,\ v = bv' \text{ for } (a,b) \in M,\ u', v' \in \Sigma^*, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then for two languages $L_1, L_2 \subseteq \Sigma^*$, we have

$$L_1 \diamond_M L_2 = \{x \diamond_M y \mid x \in L_1, y \in L_2\}.$$

The operation $\diamond_M$ is a variant of the Latin product defined in [8]. Based on $\diamond_M$, we define the *crossover operation* $\sharp_M$ for $M \subseteq \Sigma \times \Sigma$ and two languages $L_1, L_2 \subseteq \Sigma^*$ by

$$L_1 \sharp_M L_2 = \mathrm{pref}(L_1) \diamond_M \mathrm{suff}(L_2),$$

where $\mathrm{pref}(L_1)$ is the set of prefixes of words in L_1 and $\mathrm{suff}(L_2)$ is the set of suffixes of words in L_2. From this definition, the operation $\sharp_M$ can be viewed as a combination of operations under each of which the regular languages are closed. Therefore, it is easy to see that the regular languages are closed under $\sharp_M$.

Note that by restricting M to pairs (a, a) for $a \in \Sigma$, we get an operation that can be thought of as a single step of simple splicing. The operation $\sharp_M$, when restricted to pairs of the form (a, a) has some similarities to many operations that have been studied in the literature, such as the chop operation [12] and the word blending operation [5]. In fact, word blending can be seen as a special case of the crossover operation, taking $M = \{(a, a) \mid a \in \Sigma\}$.

We will now give a DFA construction for the crossover of two regular languages.

Proposition 17. *Let A and B be two DFAs defined over Σ with m and n states, respectively. Then for any $M \subseteq \Sigma \times \Sigma$, there exists a DFA C such that $L(C) = L(A)\sharp_M L(B)$ and C has at most $m \cdot 2^n$ states.*

Proof. Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be two DFAs. We will construct a DFA $C = (Q_C, \Sigma, \delta_C, s_C, F_C)$ that recognizes $A\sharp_M B$ for some $M \subseteq \Sigma \times \Sigma$, defined by

– $Q_C = Q_A \times 2^{Q_B}$,
– $s_C = \langle s_A, \emptyset \rangle$,
– $F_C = \{\langle q, P \rangle \in Q_A \times 2^{Q_B} \mid P \cap F_B \neq \emptyset\}$,

and the transition function δ_C is defined for $q \in Q_A$, $P \subseteq Q_B$, and $a \in \Sigma$ by $\delta_C(\langle q, P \rangle, a) = \langle q', P' \rangle$, where $q' = \delta_A(q, a)$ and

$$P' = \begin{cases} \mathrm{im}(\delta_B)_b & \text{if } (a, b) \in M \text{ and } q' \text{ is not a sink state,} \\ \bigcup_{p \in P} \delta_B(p, a) & \text{otherwise.} \end{cases}$$

Informally, the machine traces a computation of A and computations of B. For each pair $(a, b) \in M$, whenever a transition on a occurs in A, all states of B with incoming transitions on b are added to the computation.

It is clear from the definition of C that $L(C) = L(A) \sharp_M L(B)$ and it has at most $m \cdot 2^n$ states. □

We will show that the bound of Proposition 17 is reachable, even when M is restricted to pairs of the form (a, a).

Lemma 18. *There exist languages A_m and B_n over Σ with $|\Sigma| \geq 4$ recognized by DFAs with m and n states, respectively, and a subset $M \subseteq \Sigma \times \Sigma$ such that the minimal DFA for $L(A_m) \sharp_M L(B_n)$ requires at least $m \cdot 2^n$ states.*

The families of witness languages of Lemma 18 are accepted by DFAs A_m and B_n, shown in Figure 6, with $M = \{(d, d)\}$.

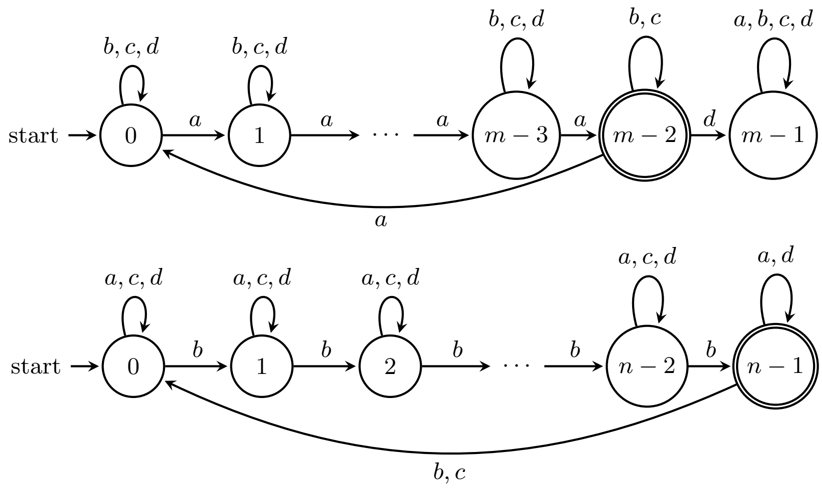

Fig. 6. The DFAs A_m (above) and B_n (below)

Together, Proposition 17 and Lemma 18 give us the following theorem.

Theorem 19. *For regular languages L_m and L_n, with $m, n \geq 3$, defined over an alphabet Σ, with $|\Sigma| \geq 4$, and a subset $M \subseteq \Sigma \times \Sigma$, if L_m has state complexity m and L_n has state complexity n, then $L_m \sharp_M L_n$ has state complexity at most $m \cdot 2^n$ and this bound can be reached in the worst case.*

6 Conclusion

We have given tight bounds for the state complexity of simple and semi-simple splicing systems and the associated crossover operation. In almost all cases, the exponential upper bound was easily reached via splicing systems defined over a fixed-size alphabet with one rule. The exception is with simple splicing systems with a finite initial language, where a natural open problem to consider is the worst-case state complexity when the initial languages are defined over alphabets of size between 3 and 2^{n-3}.

References

1. Bonizzoni, P., Ferretti, C., Mauri, G., Zizza, R.: Separating some splicing models. Inf. Process. Lett. **79**(6) (2001) 255–259
2. Câmpeanu, C., Culik II, K., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In: Automata Implementation (WIA'99), LNCS 2214 (2001) 60–70
3. Ceterchi, R.: An algebraic characterization of semi-simple splicing. Fundam. Informaticae **72** (2006) 19–25
4. Culik II, K., Harju, T.: Splicing semigroups of dominoes and DNA. Discret. Appl. Math. **31**(3) (1991) 261–277
5. Enaganti, S.K., Kari, L., Ng, T., Wang, Z.: Word blending in formal languages: The Brangelina effect. In: Unconventional Computation and Natural Computation (UCNC 2018), LNCS 10867 (2018) 72–85
6. Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. J. Autom. Lang. Comb. **21**(4) (2016) 251–310
7. Gatterdam, R.: Splicing systems and regularity. Int. J. Comput. Math. **31**(1-2) (1989) 63–67
8. Golan, J.S.: Semirings and their Applications. Springer (1999)
9. Goode, E., Pixton, D.: Semi-simple splicing systems. In: Where Math. Comput. Sci. Linguist. Biol. Meet. Springer (2001) 343–352
10. Head, T.: Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors. Bull. Math. Biol. **49**(6) (1987) 737–759
11. Head, T., Pixton, D.: Splicing and regularity. In: Recent Adv. Form. Lang. Appl. Volume 25 of Studies in Computational Intelligence. Springer (2006) 119–147
12. Holzer, M., Kutrib, M.: Descriptional and computational complexity of finite automata - A survey. Inf. Comput. **209** (2011) 456–470
13. Loos, R., Malcher, A., Wotschke, D.: Descriptional complexity of splicing systems. Int. J. Found. Comput. Sci. **19**(04) (2008) 813–826
14. Mateescu, A., Păun, G., Rozenberg, G., Salomaa, A.: Simple splicing systems. Discret. Appl. Math. **84**(1-3) (1998) 145–163
15. Pixton, D.: Regularity of splicing languages. Discret. Appl. Math. **69**(1-2) (1996) 101–124
16. Păun, G.: On the splicing operation. Discret. Appl. Math. **70**(1) (1996) 57–79

A Appendix

Here we include proofs that were omitted in the paper due to the limitation on the number of pages.

Proposition 2. *For $|\Sigma| \geq 3$ and $n \geq 3$, there exists a simple splicing system with a regular initial language $H = (\Sigma, M, L_n)$ with $|M| = 1$ where L_n is a regular language with state complexity n such that the minimal DFA for $L(H)$ requires at least $2^n - 1$ states.* (p. 6)

Proof. Let $A_n = (Q_n, \Sigma, \delta_n, 0, F_n)$ be the DFA that recognizes L_n, with $Q_n = \{0, \ldots, n-1\}$, $F_n = \{0\}$ and the transition function is defined

- $\delta(i, a) = i + 1 \bmod n$ for all $0 \leq i \leq n-1$,
- $\delta(i, b) = i$ for $0 \leq i \leq n-2$, $\delta(n-1, b) = 0$,
- $\delta(i, c) = i$ for $0 \leq i \leq n-1$.

The DFA A_n is shown in Figure 3.

Now consider the simple splicing system $H = (\Sigma, \{c\}, L_n)$. That is, H is the simple splicing system over $\Sigma = \{a, b, c\}$ with L_n as the initial language and the set of markers is $M = \{c\}$.

Consider the NFA A'_n obtained by following the construction of Proposition 1. From A'_n, we can apply the subset construction to get an equivalent DFA. Since the empty set is not reachable, there can only be at most $2^n - 1$ reachable subsets in an equivalent minimal DFA.

We will show that every nonempty subset of Q_n is reachable by showing that every nonempty subset of Q_n can be reached from Q_n. To do this, we first show that the sole subset of Q_n of size n, Q_n, is reachable from the initial state, which it is via the word c. Next, we will show that we can reach a subset of size $k-1$ from a subset of size $k > 1$. Suppose that we can reach a subset $S \subseteq Q_n$ of size k and we wish to reach the subset $S \setminus \{t\}$ for some $t \in Q_n$. There are two cases.

If $t + 1 \in S$, then we have

$$S \xrightarrow{a^{n-1-t}ba^{t+1}} S \setminus \{t\}.$$

The same argument holds for $t = n-1$ and $0 \in S$.

On the other hand, if $t + 1 \notin S$, then we must first reach state $S' = \delta'(S, a^{n-1-t})$. Observe that $t \xrightarrow{a^{n-1-t}} n-1$ and thus $n-1 \in S'$. From S', we want to reach the state $S' \setminus \{n-1\}$. Let $s = \min S'$. Then

$$S' \xrightarrow{b(a^{n-1}b)^{s-1}a^{s-1}} S' \setminus \{n-1\} \cup \{s-1\} \xrightarrow{a^{n-1-(s-1)}ba^{s}} S' \setminus \{n-1\}.$$

Finally, we shift every element of S' back to its original position in S by

$$S' \setminus \{n-1\} \xrightarrow{a^{t+1}} S \setminus \{t\}$$

and we have reached $S \setminus \{t\}$ as desired. Thus, we have shown that we can reach each subset of Q_n of size $k-1$ from a subset of Q_n of size k.

To see that each of these states is pairwise distinguishable, suppose we have two subsets S and S' with $S \neq S'$. Then without loss of generality, there is a state $t \in S$ such that $t \notin S'$ and these two states are distinguishable on the word a^{n-t}.

Thus, we have shown that a DFA recognizing $L(H)$ requires at least $2^n - 1$ states. □

(p. 6)

Proposition 4. *Let $H = (\Sigma, M, I)$ be a simple splicing system with a finite initial language, where I is a finite language recognized by a DFA A with n states. Then a DFA recognizing $L(H)$ requires at most $2^{n-2} + 1$ states.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ and let A_H be the DFA recognizing $L(H)$ obtained via the construction from Proposition 1. We will show that not all $2^n - 1$ non-empty subsets of Q are reachable in A_H. First, since I is a finite language, its DFA A is acyclic. Therefore, q_0, the initial state of A, has no incoming transitions and thus the only reachable subset containing q_0 is $\{q_0\}$. Secondly, since I is finite, A must contain a sink state, which we will call $q_\emptyset$. Note that for any subset $P \subseteq Q$, we have that P and $P \cup \{q_\emptyset\}$ are indistinguishable and can be merged together. This gives us a total of $2^{n-2} - 1 + 2$ states. □

(p. 7)

Lemma 5. *There exists a simple splicing system with a finite initial language $H = (\Sigma, M, I_n)$ where I_n is a finite language with state complexity n such that a DFA recognizing $L(H)$ requires $2^{n-2} + 1$ states.*

Proof. We can construct the DFA $A_n = (Q_n, \Sigma_n, \delta_n, 0, F_n)$ recognizing I_n, where $Q_n = \{0, \dots, n-1\}$, $\Sigma_n = \{b\} \cup \Gamma_n$ where $\Gamma_n = \{a_S \mid S \subseteq \{2, \dots, n-2\}\}$, and $F_n = \{n-2\}$. Then we define δ_n by

- $\delta_n(i, a_S) = \min\{j \in S \mid i < j \leq n-2\}$ for $1 \leq i \leq n-2$,
- $\delta_n(0, a_S) = 1$,
- $\delta_n(i, b) = i + 1$ for $0 \leq i \leq n-2$,
- $\delta_n(n-2, a) = n-1$ for all $a \in \Sigma$,
- $\delta_n(n-1, a) = n-1$ for all $a \in \Sigma$.

Then we consider the simple splicing system $H = \{\Sigma_n, \Gamma_n, I_n\}$. Let A'_n be the NFA recognizing $L(H)$ obtained via the construction from Proposition 1 and consider the DFA that results from applying the subset construction.

It is clear that by the definition of A_n that we can reach any subset $S \cup \{1\}$ with $S \subseteq \{1, \dots, n-2\}$ via the symbol a_S. Then from each of these states, we can reach a state $T = \{i_1, \dots, i_k\}$ with $2 \leq i_1 < \cdots < i_k \leq n-2\}$. If $i_1 = 2$, then we let $T' = \{i_2 - 1, \dots, i_k - 1\}$ and the subset T is reachable via the word $a_{T'}b$. If $i_1 > 2$, then the subset T is reachable via the word $a_{T' \cup \{i_1 - 1\}}b$.

To show that each of these states is pairwise distinguishable, first we note that $\{0\}$ is distinguishable from every other state by b^{n-2}. Now suppose that we have two subsets $S, S' \subseteq \{1, \dots, n-2\}$ such that $S \neq S'$. Without loss of generality, there is a state $t \in S$ such that $t \notin S'$. Then these two states can be distinguished by the word b^{n-2-t}. This gives us $2^{n-2} - 1$ states.

For the last two states, we see that $\{0\}$ is reached on the word ε and it is clearly distinguishable from every other state. The sink state $\{n-1\}$ is reachable via the word b^{n-1} and is distinguishable since it is the sole sink state of the machine. Thus, in total A'_n requires $2^{n-2}+1$ states. □

Lemma 7. *Let $H = (\Sigma, M, L)$ be a simple splicing system with a regular initial language and let A_H be an NFA recognizing $L(H)$. If $a \in M$ and δ' is the transition function of A_H, then $|\operatorname{im} \delta'_a| \leq 2$.* (p. 7)

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ be the NFA recognizing L. Let $a \in M$ and consider a state $q \in Q$. By definition of A'_H, if $\delta(q,a) = \{n-1\}$, then $\delta'(q,a) = \{n-1\}$. Otherwise $\delta'(q,a) = \operatorname{im} \delta_a$. Since these are the only two possibilities, we have $|\operatorname{im} \delta'_a| \leq 2$. □

Proposition 8. *Let $H = (\{a\}, M, I)$ be a simple splicing system where I is a finite language containing a word of length at least 2. Then the minimal DFA recognizing $L(H)$ has exactly two states.* (p. 7)

Proof. Since the alphabet is $\{a\}$, if M is nonempty, we have $M = \{a\}$ (otherwise, if $M = \emptyset$, then $L(H) = I$). If I does not contain a word of length at least 2, then either $I = \{\varepsilon\}$ or $I = \{a\}$. Then, it is clear that for any finite language I with $w \in I$ such that $|w| \geq 2$, we have $L(H) = a^+$. Thus, a DFA recognizing $L(H)$ has exactly two states. □

Proposition 9. *Let $H = (\{a,b\}, M, I)$ be a simple splicing system where I is a finite language with state complexity n. Then the minimal DFA recognizing $L(H)$ has at most $2n-3$ states and this bound is reachable in the worst case.* (p. 7)

Proof. Recall from Lemma 7 that the action of a symbol $c \in M$ has an image of size 2, containing $\operatorname{im} \delta_c \subseteq \{1, \ldots, n-2\}$ and $\{n-1\}$. In order to maximize the number of states of A_H, we must have $a \notin M$ and $b \in M$. Furthermore, δ_a must be the action $i \mapsto i+1$ for $0 \leq i \leq n-2$. Then there are n subsets of size 1 and up to $n-3$ subsets of size $|\operatorname{im} \delta_b| \geq 2$. This gives at most $2n-3$ states.

We will show that this bound is reachable. Let $A_n = (Q_n, \{a,b\}, \delta_n, 0, \{n-2\})$ be a DFA, with $Q_n = \{0, \ldots, n-1\}$ and δ_n is defined by

- $\delta_n(i,a) = i+1$ for $0 \leq i \leq n-2$,
- $\delta_n(0,b) = 1$, $\delta_n(1,b) = 2$, $\delta_n(i,b) = n-1$ for $2 \leq i \leq n-2$,
- $\delta_n(n-1,d) = n-1$ for all $d \in \{a,b\}$.

The DFA A_n is shown in Figure 7.

Now, we consider the splicing system $H = (\{a,b\}, \{b\}, L(A_n))$ and let A'_n be the DFA obtained by the construction from Proposition 1. We claim that the reachable states of Q'_n are either of the form $\{i\}$ for $i \in Q_n$ or $\{i, i+1\}$, for $i \in \{1, \ldots, n-3\}$. We will show that each of these states is reachable.

To reach states of the form $\{i\}$ for $1 \leq i \leq n-1$, we have $\{0\} \xrightarrow{a^i} \{i\}$. The state $\{1,2\}$ is reached from the initial state via the word b. Then from

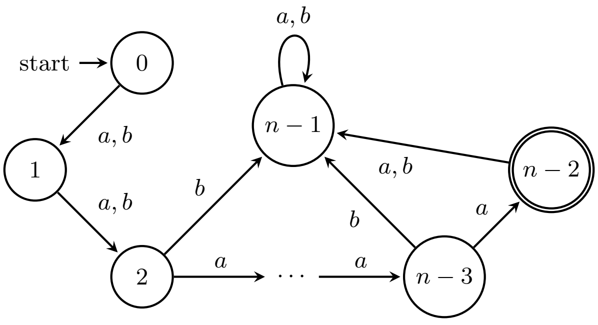

Fig. 7. The binary witness DFA A_n

the state $\{1,2\}$, we can reach states of the form $\{i, i+1\}$ for $2 \le i \le n-3$ by $\{1,2\} \xrightarrow{a^{i-1}} \{i, i+1\}$. To see that these states are pairwise distinguishable, consider two subsets S and S' of Q_n with $S \neq S'$. Then there is some $t \in S$ such that $t \notin S'$. Then the two states are distinguished on the word a^{n-2-t}.

Thus A'_n has $n - 3 + n = 2n - 3$ states that are reachable and pairwise distinguishable. □

(p. 7) **Proposition 10.** *Let $H = (\{a,b,c\}, M, I)$ be a simple splicing system where I is a finite language with state complexity n. Then the minimal DFA recognizing $L(H)$ has at most $2^{\frac{n}{2}} + 1$ states if n is even and $3 \cdot 2^{\frac{n-3}{2}} + 1$ states if n is odd.*

Proof. Let $A = (Q, \{a,b,c\}, \delta, 0, F)$ be the minimal DFA that recognizes I and let A_H be the DFA obtained by the construction from Proposition 1. We claim that in order to maximize the number of states of A_H, we must have $c \in M$ and $a, b \notin M$. Recall from Lemma 7 that if $c \in M$, then $|\operatorname{im} \delta'_c| = 2$. Observe that if $M = \{b, c\}$, then δ_a must be the action $i \mapsto i+1$, which gives at most $3(n-2)+1$ states. Thus, it must be the case that both $a, b \notin M$.

Consider the sets of states that are reached via words in $c\cdot(\Sigma \setminus M)^* = c\cdot\{a,b\}^*$. Consider one such word cw where $w \in \{a,b\}^*$. We say a state $q \in Q$ is in level i with respect to c if $q \in \operatorname{im} \delta'_{cw}$ and $|w| = i$. For example, every state in the set $\operatorname{im} \delta_c$ is in level 0 with respect to c.

Recall that a DFA for a finite language is acyclic and its states are ordered. Since I is finite, there is at least one state of A that is in level i and is not in level $i+1$. That is, at each step of the computation of a word cw, where $w \in \{a,b\}^*$, when reading symbols in $\{a,b\}$, there is at least one state that becomes unreachable because the original DFA A is acyclic. However, we can "reset" the set of reachable subsets by reading a symbol in M, in this case c, and we "reset" our computation to the set of states $\operatorname{im} \delta_c$.

This gives a bound on the number of subsets of states that are reachable in A_H. On level i with respect to c, there are at most 2^{n-2-i} reachable subsets of states. However, the number of subsets is also bound by the number of words that can reach each subset. Thus, there are at most $|\Sigma \setminus M|^i$ subsets of states which are reachable. The number of reachable subsets is thus bounded by

$$\sum_{i=0}^{n-3} \min\{2^{n-2-i}, |\Sigma \setminus M|^i\} = \sum_{i=0}^{t-1} |\Sigma \setminus M|^i + 2^{n-2-t},$$

where $t = \min\{i \in \mathbb{N} \mid 2^{n-2-i} \leq |\Sigma \setminus M|^i\}$. For $|\Sigma| = 3$ and $|M| = 1$, this gives us $t = \frac{n-2}{2}$ if n is even and $t = \frac{n-1}{2}$ if n is odd. Thus, for n even, there are at most

$$\sum_{i=0}^{\frac{n-2}{2}-1} 2^i + 2^{n-2-\frac{n-2}{2}} + 2 = 2^{\frac{n-2}{2}} - 1 + 2^{\frac{n-2}{2}} + 2 = 2^{\frac{n}{2}} + 1$$

states in A' and for n odd, there are at most

$$\sum_{i=0}^{\frac{n-1}{2}-1} 2^i + 2^{n-2-\frac{n-1}{2}} + 2 = 2^{\frac{n-1}{2}} - 1 + 2^{\frac{n-3}{2}} + 2 = 3 \cdot 2^{\frac{n-3}{2}} + 1$$

states in A'. □

Lemma 11. *There exists a family of finite languages $I_n \subseteq \{a, b, c\}^*$, for $n \geq 4$, recognized by a DFA with n states such that the minimal DFA for a simple splicing system $H = (\{a, b, c\}, M, I_n)$ requires at least $2^{\frac{n}{2}} + 1$ states if n is even and $3 \cdot 2^{\frac{n-3}{2}} + 1$ states if n is odd.* (p. 8)

Proof. Let I_n be recognized by the DFA $A_n = (Q_n, \{a, b, c\}, \delta_n, 0, \{n-2\})$, with $Q_n = \{0, \ldots, n-1\}$ and where δ_n is defined by

- $\delta_n(i, a) = i + 1$ for $0 \leq i \leq n - 2$,
- $\delta_n(i, b) = i + 1$ for $0 \leq i \leq \lfloor \frac{n}{2} \rfloor - 2$ and $\lfloor \frac{n}{2} \rfloor \leq i \leq n - 2$,
- $\delta_n(\lfloor \frac{n}{2} \rfloor - 1, b) = n - 1$,
- $\delta_n(i, c) = i + 1$ for $0 \leq i \leq \lfloor \frac{n}{2} \rfloor - 2$,
- $\delta_n(i, c) = n - 1$ for $\lfloor \frac{n}{2} \rfloor - 1 \leq i \leq n - 1$,
- $\delta_n(n - 1, d) = n - 1$ for all $d \in \Sigma$.

The DFA A_n is shown in Figure 4.

We obtain a DFA A'_n recognizing $L(H)$ by performing the construction from Proposition 1 on the DFA A_n and applying the subset construction to the resultant NFA. We will consider the number of reachable and pairwise distinguishable states of A'_n.

First, we consider the reachable states of A'_n. Let $S_i \subseteq \{1, \ldots, n-2\}$ for $1 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1$. We will show that states of the form $S_i = \{i+1, i+2, \ldots, \lfloor \frac{n}{2} \rfloor - 1\} \cup P_i$, where $P_i \subseteq \{\lfloor \frac{n}{2} \rfloor, \ldots, \lfloor \frac{n}{2} \rfloor + i\}$ are reachable on words uv where $u \in \Sigma^*$ and $v \in c\{a, b\}^i$.

For $i = 0$, we have $uv = uc$ on which the subset $\operatorname{im} \delta_c = \{1, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$ is reached. Now consider $i > 0$ and let $P_i = \{j_1, j_2, \ldots, j_k\} \subseteq \{\lfloor \frac{n}{2} \rfloor, \ldots, \lfloor \frac{n}{2} \rfloor - 1 + i\}$

for $k \leq i$. The state $S_i = \{i+1, \ldots, \lfloor \frac{n}{2} \rfloor - 1\} \cup P_i$ is reachable on the word $uv = uca_1a_2 \cdots a_i$, where for $1 \leq j \leq i$,

$$a_j = \begin{cases} a & \text{if } \lfloor \frac{n}{2} \rfloor - 1 + j \in P_i \\ b & \text{otherwise.} \end{cases}$$

Then for each i, there are 2^i reachable states for $0 \leq i \leq \lfloor \frac{n-2}{2} \rfloor$. If n is even, this gives us $2^{\frac{n-2}{2}+1} - 1$ states that can be reached. Together with the initial and sink states, this gives a total of $2^{\frac{n}{2}} + 1$ states. If n is odd this gives a total of $3 \cdot 2^{\frac{n-3}{2}} + 1$ states that can be reached.

To show that each of these states is pairwise distinguishable, consider two states $S, T \subseteq \{1, \ldots, n-2\}$. If $S \neq T$, then there exists some element $q \in S$ such that $q \notin T$ and S and T are distinguishable on the word a^{n-2-q}. Finally, it is clear that $\{0\}$ and $\{n-1\}$ are distinguishable from any state $S \subseteq \{1, \ldots, n-2\}$.

Thus, we have shown that A'_n has $2^{\frac{n}{2}}+1$ reachable and pairwise distinguishable states if n is even and $3 \cdot 2^{\frac{n-3}{2}} + 1$ reachable and pairwise distinguishable states if n is odd. □

(p. 9) **Proposition 13.** *Let $H = (\Sigma, M^{(2)}, L)$ be a semi-simple splicing system with a regular initial language. Then there exists an NFA B'_H with n states such that $L(B'_H) = L(H)$.*

Proof. Let $H = (\Sigma, M^{(2)}, L)$ and let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. We will define the NFA $B_H = (Q', \Sigma, \delta', q_0, F)$, where $Q' = Q \cup Q_M$ with $Q_M = \{p_a, p_b \mid (a, b) \in M^{(2)}\}$ and the transition function δ' is defined

- $\delta'(q, a) = \{\delta(q, a)\}$ if $q \in Q$ and $a \in \Sigma$,
- $\delta'(q, \varepsilon) = \{p_a\}$ if $q \in Q$, $a \in M$, and $\delta(q, a)$ is not the sink state,
- $\delta'(p_a, a) = \{p_b\}$ if $p_a \in Q_M$ and $(a, b) \in M^{(2)}$,
- $\delta'(p_b, \varepsilon) = \operatorname{im} \delta_b$ if $p_b \in Q_M$ and $(a, b) \in M^{(2)}$ for some $a \in \Sigma$.

We will describe the construction briefly, as it is similar to the construction described in Proposition 1. Recall that each marker (a, b) in $M^{(2)}$ corresponds to a splicing rule $(a, \varepsilon; b, \varepsilon)$. For each such rule, add states p_a and p_b and add a transition $p_a \xrightarrow{a} p_b$. For every state $q \in Q$ with outgoing transitions on a, add ε-transitions to p_a and add ε-transitions from p_b to all states with incoming transitions on b. Recall that $\operatorname{im} \delta_b$ is the image of the transformation of δ induced by b, and therefore it is the set of states of A with an incoming transition on b.

We can now simplify this NFA by removing ε-transitions in the usual way to obtain an NFA $B'_H = (Q, \Sigma, \delta', q_0, F)$, where

$$\delta'(q, a) = \begin{cases} \{\delta(q, a)\} \cup \operatorname{im} \delta_b & \text{if } (a, b) \in M^{(2)}, \\ \{\delta(q, a)\} & \text{otherwise.} \end{cases}$$

Similar to the construction for NFAs recognizing the language of simple splicing systems from Proposition 1, observe that by removing the ε-transitions, we also remove the states that were initially added earlier in the construction of B_H. Thus, the state set of B'_H is exactly the state set of the DFA A recognizing L. □

Lemma 15. *Let $n \geq 4$. Then there exists a semi-simple splicing system with a finite initial language $H = (\Sigma, M^{(2)}, I_n)$ where $|\Sigma| = 3$ and I_n is a finite language with state complexity n such that $L(H)$ is recognized by a DFA that requires at least $2^{n-2} + 1$ states.* (p. 9)

Proof. Let I_n be recognized by the DFA $A_n = (Q_n, \{a, b, c\}, \delta_n, 0, F_n)$, where $Q_n = \{0, \dots, n-1\}$ and $F_n = \{n-2\}$. We define δ_n by

- $\delta_n(i, a) = i + 1$ for $0 \leq i \leq n-2$,
- $\delta_n(i, b) = i + 1$ for $0 \leq i \leq n-2$,
- $\delta_n(0, c) = 1$, $\delta_n(i, c) = n-1$ for $1 \leq i \leq n-2$,
- $\delta_n(n-1, d) = n-1$ for all $d \in \Sigma$.

The DFA A_n is shown in Figure 5.

We define the semi-simple splicing system $H = (\{a, b, c\}, M^{(2)}, I_n)$ with $M^{(2)} = \{(a, c)\}$ and let B'_n be the NFA recognizing $L(H)$ obtained via the construction of Proposition 13.

It is clear that the initial state $\{0\}$ and the sink state $\{n-1\}$ are reachable. We will then show that all states $S \subseteq \{1, \dots, n-2\}$ are reachable. Let $S = \{s_1, \dots, s_k\} \subseteq \{1, \dots, n-2\}$ with $1 \leq s_1 < \cdots < s_k \leq n-2$. Then

$$\delta'_n(S, d) = \begin{cases} \{1, s_1 + 1, \dots, s_k + 1\} & \text{if } d = a, \\ \{s_1 + 1, \dots, s_k + 1\} & \text{if } d = b, \\ \{n-1\} & \text{if } d = c, \end{cases}$$

Then each subset S is reachable from the initial state $\{0\}$ via the word $w = x_1 x_2 \cdots x_{s_k}$ where

$$x_i = \begin{cases} a & \text{if } s_k - i + 1 \in S, \\ b & \text{if } s_k - i + 1 \notin S. \end{cases}$$

Now we show that each of these states is pairwise distinguishable. Consider two subsets $S, S' \subseteq \{1, \dots, n-2\}$ with $S \neq S'$. Without loss of generality, let $t \in S$ such that $t \notin S'$. Then S and S' are distinguishable via the word b^{n-2-t}. Thus, we have shown that every nonempty subset of $\{1, \dots, n-2\}$ is reachable and pairwise distinguishable and there are $2^{n-2} - 1$ such subsets.

Together with the initial state $\{0\}$ and the sink state $\{n-1\}$, we have shown that B'_n has $2^{n-2} + 1$ reachable and pairwise distinguishable states. □

Lemma 18. *There exist languages A_m and B_n over Σ with $|\Sigma| \geq 4$ recognized by DFAs with m and n states, respectively, and a subset $M \subseteq \Sigma \times \Sigma$ such that the minimal DFA for $L(A_m) \sharp_M L(B_n)$ requires at least $m \cdot 2^n$ states.* (p. 11)

Proof. Let $\Sigma = \{a, b, c, d\}$ and let $M = \{(d, d)\}$. Let A_m be recognized by the DFA $A_m = (Q_A, \Sigma, \delta_A, s_A, F_A)$, where $Q_A = \{0, \dots, m-1\}$, $s_A = 0$, $F_A = \{m-2\}$, and the transition function δ_A is defined by

- $\delta_A(i, a) = i + 1 \bmod m - 1$ for $0 \leq i \leq m-2$,

- $\delta_A(i, b) = i$ for $0 \le i \le m-2$,
- $\delta_A(i, c) = i$ for $0 \le i \le m-2$,
- $\delta_A(i, d) = i$ for $0 \le i \le m-3$, $\delta_A(m-2, d) = m-1$.
- $\delta_A(m-1, \sigma) = m-1$ for all $\sigma \in \Sigma$.

Note that the state $m-1$ acts as the sink state of A_m.

Let B_n be recognized by the DFA $B_n = (Q_B, \Sigma, \delta_B, s_B, F_B)$, where $Q_B = \{0, \ldots, n-1\}$, $s_B = 0$, $F_B = \{n-1\}$, and the transition function δ_B is defined by

- $\delta_B(i, a) = i$ for $0 \le i \le n-1$,
- $\delta_B(i, b) = i + 1 \bmod n$ for $0 \le i \le n-1$,
- $\delta_B(i, c) = i$ for $0 \le i \le n-2$, $\delta_B(n-1, c) = 0$,
- $\delta_B(i, d) = i$ for $0 \le i \le n-1$.

Observe that B_n has no sink state. The DFAs A_m and B_n are shown in Figure 6.

Consider the DFA C' obtained by applying the construction from Proposition 17 on A_m and B_n and taking $M = \{(d, d)\}$. We will show that every state in $Q_A \times 2^{Q_B}$ is reachable.

First, $\langle 0, \emptyset \rangle$ is reachable since it is the initial state. Then we can show that the state $\langle q, \emptyset \rangle$ is reachable for each $1 \le q \le m-2$ by $\langle 0, \emptyset \rangle \xrightarrow{a^q} \langle q, \emptyset \rangle$. Finally, $\langle m-1, \emptyset \rangle$ is reachable from $\langle m-2, \emptyset \rangle$ on the word d.

Next, we will show how to reach every state $\langle q, S \rangle$ for $q \in Q_A$ and $S \subseteq Q_B$. We will first show that each state $\langle q, Q_B \rangle$, $0 \le q \le m-3$, is reachable from $\langle q, \emptyset \rangle$ by reading d. Then $\langle m-3, Q_B \rangle \xrightarrow{a} \langle m-2, Q_B \rangle$ and $\langle m-2, Q_B \rangle \xrightarrow{d} \langle m-1, Q_B \rangle$. We can then show that for each subset $S \subseteq Q_B$, the state $\langle q, S \rangle$ is reachable by the approach used in the proof of Proposition 2. We can do this by using words over $\{b, c\}$, which keeps the first component of the state fixed.

Now we will see that each of these states is pairwise distinguishable. Suppose we have two states $\langle q, S \rangle$ and $\langle q', S' \rangle$. First, suppose that $S \neq S'$ and that there is an element $t \in S$ with $t \notin S'$. Then $\langle q, S \rangle$ and $\langle q', S' \rangle$ are distinguishable via the word b^{n-1-t}.

Now suppose that $S = S'$ but $q \neq q'$ and without loss of generality, $q < q'$. There are two cases. First, if $S = Q_B$, then $\langle q, S \rangle \xrightarrow{c} \langle q, Q_B \setminus \{n-1\} \rangle$ and let $T = Q_B \setminus \{n-1\}$. If $S \neq Q_B$, then let $t = \max(Q_B \setminus S)$ and denote by $T \subseteq Q_B$ the subset such that $\langle q, S \rangle \xrightarrow{b^{n-1-t}} \langle q, T \rangle$. In either case, we have $T \subseteq Q_B \setminus \{n-1\}$ and we can consider states $\langle q, T \rangle$ and $\langle q', T \rangle$ obtained via the same words.

Then to distinguish $\langle q, T \rangle$ and $\langle q', T \rangle$, first suppose that $q < q' \le m-2$. This gives us

$$\langle q, T \rangle \xrightarrow{a^{m-2-q'}d} \langle q + (m-2-q'), Q_B \rangle \text{ and } \langle q', T \rangle \xrightarrow{a^{m-2-q'}d} \langle m-1, T \rangle,$$

which puts us in the above case when $S \neq S'$. Next, if $q' = m-1$ and $q < m-2$, then we can enter the same situation via the word d. Finally, if $q' = m-1$ and $q = m-2$, we can enter the same scenario via the word ad.

Thus, we have shown that all $m \cdot 2^n$ states are reachable and pairwise distinguishable, and thus C' requires at least $m \cdot 2^n$ states. $\square$