Descriptional Complexity of Semi-simple Splicing Systems

Lila Kari[1] and Timothy Ng[1,2(✉)]

[1] School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada
lila.kari@uwaterloo.ca, timng@uchicago.edu
[2] Department of Computer Science, University of Chicago, Chicago, IL 60637, USA

Abstract. Splicing systems are generative mechanisms introduced by Tom Head in 1987 to model the biological process of DNA recombination. The computational engine of a splicing system is the "splicing operation", a cut-and-paste binary string operation defined by a set of "splicing rules", quadruples $r = (u_1, u_2; u_3, u_4)$ where u_1, u_2, u_3, u_4 are words over an alphabet Σ. For two strings $x_1u_1u_2y_1$ and $x_2u_3u_4y_2$, applying the splicing rule r produces the string $x_1u_1u_4y_2$. In this paper we focus on a particular type of splicing systems, called (i, j) semi-simple splicing systems, $i = 1, 2$ and $j = 3, 4$, wherein all splicing rules r have the property that the two strings in positions i and j in r are singleton letters, while the other two strings are empty. The language generated by such a system consists of the set of words that are obtained starting from an initial set called "axiom set", by iteratively applying the splicing rules to strings in the axiom set as well as to intermediately produced strings. We consider semi-simple splicing systems where the axiom set is a regular language, and investigate the descriptional complexity of such systems in terms of the size of the minimal deterministic finite automata that recognize the languages they generate.

1 Introduction

Splicing systems are generative mechanisms introduced by Tom Head [7] to model the biological process of DNA recombination. A splicing system consists of an initial language called an *axiom set*, and a set of so-called *splicing rules*. The result of applying a splicing rule to a pair of operand strings is a new "recombinant" string, and the language generated by a splicing system consists of all the words that can be obtained by successively applying splicing 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: the class of languages generated by finite Head splicing systems [7] is strictly contained in the class of languages generated

N. Jonoska and D. Savchuk (Eds.): DLT 2020, LNCS 12086, pp. 150–163, 2020.
https://doi.org/10.1007/978-3-030-48516-0_12

by finite Păun splicing systems [13], which is strictly contained in the class of languages generated by finite Pixton splicing systems [12].

In this paper we will use the Păun definition [13], which defines a splicing rule as a quadruplet of words $r = (u_1, u_2; u_3, u_4)$. This rule splices two words $x_1u_1u_2y_1$ and $x_2u_3u_4y_2$ as follows: The words are cut between the factors u_1, u_2, respectively u_3, u_4, and the prefix of the first word (ending in u_1) is recombined by catenation with the suffix of the second word (starting with u_4), resulting in the word $x_1u_1u_4y_2$.

Culik II and Harju [3] proved that finite Head splicing systems can only generate regular languages, while [8] and [12] proved a similar result for Păun, respectively Pixton splicing systems. Gatterdam [5] gave $(aa)^*$ as an example of a regular language which cannot be generated by a finite Head splicing system, which proved that this is a strict inclusion.

As the classes of languages generated by finite splicing systems are subclasses of the family of regular languages, their descriptional complexity can be considered in terms of the finite automata that recognize them. For example, Loos et al. [10] gave a bound on the number of states required for a nondeterministic finite automaton to recognize the language generated by an equivalent Păun finite splicing system. Other descriptional complexity measures for finite splicing systems that have been investigated in the literature include the number of rules, the number of words in the initial language, the maximum length of a word in the initial axiom set, and the sum of the lengths of all words in the axiom set. Păun [13] also proposed the radius, defined to be the size of the largest u_i in a rule, as another possible measure.

In the original definition, simple splicing systems are finite splicing systems where all the words in the splicing rules are singleton letters. The descriptional complexity of simple splicing systems was considered by Mateescu et al. [11] in terms of the size of a right linear grammar that generates a simple splicing language. Semi-simple splicing systems were introduced in Goode and Pixton [6] as having a finite axiom set, and splicing rules of the form $(a, \varepsilon; b, \varepsilon)$ where a, b are singleton letters, and ε denotes the empty word.

In this paper we focus our study on some variants of semi-simple splicing systems called (i, j)-semi-simple splicing systems, $i = 1, 2$ and $j = 3, 4$, wherein all splicing rules have the property that the two strings in positions i and j are singleton letters, while the other two strings are empty. (Note that Ceterchi et al. [2] showed that all classes of languages generated by semi-simple splicing systems are pairwise incomparable[1]). In addition, in a departure from the original definition of semi-simple splicing systems [6], in this paper the axiom set is allowed to be a (potentially infinite) regular set.

More precisely, we investigate the descriptional complexity of (i, j)-semi-simple splicing systems with regular axiom sets, in terms of the size of the minimal deterministic finite automaton that recognizes the language generated by the system. The paper is organized as follows: Sect. 2 introduces definitions

[1] Simple splicing language classes are pairwise incomparable except for the pair (1,3) and (2,4), which are equivalent [11].

and notations, Sect. 3 defines splicing systems and outlines some basic results on simple splicing systems, Sects. 4, 5 and 6 investigate the state complexity of (2,4)-, (2,3)- respectively (1,4)-semi-simple splicing systems, and Sect. 7 summarizes our results (Table 1).

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 $\delta : Q \times \Sigma \to Q$, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ 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 $a \in \Sigma$. In this paper, all DFAs are defined to be complete. We will also make use of the notation $q \xrightarrow{w} q'$ for $\delta(q, w) = q'$, where $w \in \Sigma^*$ and $q, q' \in Q$. The language recognized or accepted by A is $L(A) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$.

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 $\operatorname{im} \delta_a = \{\delta(p, a) \mid p \in Q\}$.

A state q is called *reachable* if there exists a string $w \in \Sigma^*$ such that $\delta(q_0, w) = q$. A state q is called *useful* if there exists a string $w \in \Sigma^*$ such that $\delta(q, w) \in F$. A state that is not useful is called *useless*. A complete DFA with multiple useless states can be easily transformed into an equivalent DFA with at most one useless state, which we refer to as the *sink state*.

Two states p and q of A are said to be *equivalent* or *indistinguishable* in the case that $\delta(p, w) \in F$ if and only if $\delta(q, w) \in F$ for every word $w \in \Sigma^*$. States that are not equivalent are *distinguishable*. 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 [4].

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 $F \subseteq Q$ 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 $\delta_a(q) = \delta(q, a)$. We define $\operatorname{im} \delta_a = \bigcup_{q \in Q} \delta_a(q)$. We make use of the notation $P \xrightarrow{w} P'$ for $P' = \bigcup_{q \in P} \delta(q, w)$, where $w \in \Sigma^*$ and $P, P' \subseteq Q$.

3 Semi-simple Splicing Systems

In this paper we will use the notation of Păun [13]. The splicing operation is defined via sets of quadruples $r = (u_1, u_2; u_3, u_4)$ with $u_1, u_2, u_3, u_4 \in \Sigma^*$ called splicing rules. For two strings $x = x_1u_1u_2x_2$ and $y = y_1u_3u_4y_2$, applying the rule $r = (u_1, u_2; u_3, u_4)$ produces a string $z = x_1u_1u_4y_2$, 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 $L(H) = \sigma^*(L)$.

Goode and Pixton [6] define a restricted class of splicing systems called semi-simple splicing systems. A semi-simple splicing system is a triple $H = (\Sigma, M, I)$, where Σ is an alphabet, $M \subseteq \Sigma \times \Sigma$ is a set of markers, and I is a finite initial language over Σ. We have $(x, y) \vdash^{(a,b)} z$ if and only if $x = x_1ax_2$, $y = y_1by_2$, and $z = x_1ay_2$ for some $x_1, x_2, y_1, y_2 \in \Sigma^*$. That is, a semi-simple splicing system is a splicing system in which the set of rules is $\mathcal{M} = \{(a, \varepsilon; b, \varepsilon) \mid (a, b) \in M\}$. Since the rules are determined solely by our choice of $M \subseteq \Sigma \times \Sigma$, the set M is used in the definition of the semi-simple splicing system rather than the set of rules $\mathcal{M}$.

It is shown in [6] that the class of languages generated by semi-simple splicing systems is a subclass of the regular languages. Semi-simple splicing systems are a generalization of the class of simple splicing systems, defined by Mateescu et al. [11]. A splicing system is a simple splicing system if it is a semi-simple splicing system and all markers are of the form (a, a) for $a \in \Sigma$. It is shown in [11] that the class of languages generated by simple splicing systems is a subclass of the extended star-free languages.

Observe that the set of rules $\mathcal{M} = \{(a, \varepsilon; b, \varepsilon) \mid (a, b) \in M\}$ of a semi-simple splicing system consist of 4-tuples with symbols from Σ in positions 1 and 3 and ε in positions 2 and 4. We can call such splicing rules (1,3)-splicing rules. Then a (1,3)-splicing system is a splicing system with only (1,3)-splicing rules and ordinary semi-simple splicing systems can be considered (1,3)-semi-simple splicing systems. The state complexity of (1,3)-simple and (1,3)-semi-simple splicing systems was studied previously by the authors in [9].

We can consider variants of semi-simple splicing systems in this way by defining semi-simple (i, j)-splicing systems, for $i = 1, 2$ and $j = 3, 4$. A semi-simple (2,4)-splicing system is a splicing system (Σ, M, I) with rules $\mathcal{M} = \{(\varepsilon, a; \varepsilon, b) \mid$

$(a, b) \in M\}$. A (2,3)-semi-simple splicing system is a splicing system (Σ, M, I) with rules $\mathcal{M} = \{(\varepsilon, a; b, \varepsilon) \mid (a, b) \in M\}$. A (1,4)-semi-simple splicing system is a semi-simple splicing system (Σ, M, I) with rules $\mathcal{M} = \{(a, \varepsilon; \varepsilon, b) \mid (a, b) \in M\}$.

The classes of languages generated by simple and semi-simple splicing systems and their variants have different relationships among each other. Mateescu et al. [11] show that the classes of languages generated by (1,3)-simple splicing systems (i.e. ordinary simple splicing systems) and (2,4)-simple splicing systems are equivalent, while, the classes of languages generated by (1,3)-, (1,4)-, and (2,3)-simple splicing systems are all incomparable and subregular.

The situation is different for semi-simple splicing systems. Ceterchi et al. [2] show that each of the classes of languages generated by (1,3)-, (1,4)-, (2,3)-, and (2,4)-semi-simple splicing systems are all incomparable. So unlike simple splicing systems, the (1,3)- and (2,4)- variants are *not* equivalent. They show this by showing that the language $a^+ \cup a^+ab \cup aba^+ \cup aba^+b$ is generated by the (1,3)-semi-simple splicing system $(\{a, b\}, \{(a, \varepsilon; b, \varepsilon)\}, \{abab\})$ but cannot be generated by a (2,4)-semi-simple splicing system, while the language $b^+ \cup abb^+ \cup b^+ab \cup ab^+ab$ can be generated by the (2,4)-semi-simple splicing system $(\{a, b\}, \{(\varepsilon, a; \varepsilon, b)\}, \{abab\})$ but not a (1,3)-semi-simple splicing system.

In this paper, we will relax the condition that the initial language of a semi-simple splicing system must be a finite language, and we will consider also semi-simple splicing systems with regular initial languages. By [13], 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.

4 State Complexity of (2,4)-semi-simple Splicing Systems

In this section, we will consider the state complexity of (2,4)-semi-simple splicing systems. Recall that a (2,4)-semi-simple splicing system is a splicing system with rules of the form $(\varepsilon, a; \varepsilon, b)$ for $a, b \in \Sigma$. As mentioned previously, the classes of languages generated by (1,3)- and (2,4)-simple splicing systems were shown to be equivalent by Mateescu et al. [11], while the classes of languages generated by (1,3)- and (2,4)-semi-simple splicing systems were shown to be incomparable by Ceterchi et al. [2].

First, we define an NFA that recognizes the language of a given (2,4)-semi-simple splicing system. This construction is based on the construction of Head and Pixton [8] for Păun splicing rules, which is based on the construction for Pixton splicing rules by Pixton [12]. The original proof of regularity of finite splicing is due to Culik and Harju [3]. We follow the Head and Pixton construction and apply ε-transition removal on the resulting NFA to obtain an NFA for the semi-simple splicing system with the same number of states as the DFA for the initial language of the splicing system.

Proposition 1. *Let $H = (\Sigma, M, L)$ be a (2,4)-semi-simple splicing system with a regular initial language and let L be recognized by a DFA with n states. Then there exists an NFA A'_H with n states such that $L(A'_H) = L(H)$.*

The result of this construction is an NFA that "guesses" when a splicing operation occurs. Since each component of a semi-simple splicing rule is of length at most 1, the construction of the NFA need only consider the outgoing and incoming transitions of states. In the case of (2,4)-semi-simple splicing systems, for a rule (a, b), any state with an outgoing transition on a has added transitions on a to every state with an incoming transition on b.

From this NFA construction, we can obtain a DFA via subset construction. This gives an upper bound of $2^n - 1$ reachable states. This upper bound is the same for (1,3)-simple and (1,3)-semi-simple splicing systems and was shown to be tight [9]. Since (1,3)-simple splicing systems and (2,4)-simple splicing systems are equivalent, we state without proof that the same result holds for (2,4)-simple splicing systems via the same lower bound witness. Therefore, this bound is reachable for (2,4)-semi-simple splicing systems via the same lower bound witness.

Proposition 2 [9]. *For $|\Sigma| \geq 3$ and $n \geq 3$, there exists a (2,4)-simple splicing system with a regular initial language $H = (\Sigma, M, L)$ with $|M| = 1$ where L is a regular language with state complexity n such that the minimal DFA for $L(H)$ requires at least $2^n - 1$ states.*

It was also shown in [9] that if the initial language is finite, this upper bound is not reachable for (1,3)-simple and (1,3)-semi-simple splicing systems. This result holds for all variants of semi-simple splicing systems and the proof is exactly the same as in [9]. We state the result for semi-simple splicing systems for completeness.

Proposition 3 [9]. *Let $H = (\Sigma, M, I)$ 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.*

This upper bound is witnessed by a (2,4)-semi-simple splicing system which requires both an alphabet and ruleset that grows exponentially with the number of states of the initial language. This is in contrast to the lower bound witness for (1,3)-semi-simple systems from [9], which requires only three letters. We also note that the initial language used for this witness is the same as that for (1,3)-simple splicing systems from [9]. From this, we observe that the choice of the visible sites for the splicing rules (i.e. (1,3) vs. (2,4)) makes a difference in the state complexity. We will see other examples of this later as we consider semi-simple splicing systems with other rule variants.

Theorem 4. *Let $H = (\Sigma, M, I)$ be a (2,4)-semi-simple splicing system with a finite initial language, where I is a finite language with state complexity n and $M \subseteq \Sigma \times \Sigma$. Then 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 (2,3)-semi-simple Splicing Systems

We will now consider the state complexity of (2,3)-semi-simple splicing systems. Recall that a (2,3)-semi-simple splicing system is a splicing system with rules

of the form $(\varepsilon, a; b, \varepsilon)$ for $a, b \in \Sigma$. We can follow the same construction from Proposition 1 with slight modifications to account for $(2,3)$-semi-simple splicing rules to obtain an NFA for a language generated by a (2,3)-semi-simple splicing system with the same number of states as the DFA for the initial language of the splicing system.

Proposition 5. *Let $H = (\Sigma, M, L)$ be a (2,3)-semi-simple splicing system with a regular initial language and let L be recognized by a DFA with n states. Then there exists an NFA A'_H with n states such that $L(A'_H) = L(H)$.*

Note that in this NFA construction, for each (2,3)-semi-simple splicing rule (a, b), any state with an outgoing transition on a has additional ε-transitions to every state with an incoming transition on b. This differs from the NFA construction for (2,4)-semi-simple splicing systems, where the new transitions were on the symbol a. From this NFA, we then get an upper bound of $2^n - 1$ reachable states via the subset construction. However, we will show that because of the ε-transitions, this bound cannot be reached.

Proposition 6. *Let $H = (\Sigma, M, L)$ be a (2,3)-semi-simple splicing system with a regular initial language, where $M \subseteq \Sigma \times \Sigma$ and $L \subseteq \Sigma^*$ is recognized by a DFA with n states. Then there exists a DFA A_H such that $L(A_H) = L(H)$ and A_H has at most 2^{n-1} states.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ be the DFA for L and let $B_H = (Q, \Sigma, \delta', q_0, F)$ be the NFA obtained via the construction of Proposition 5 given the (2,3)-semi-simple splicing system H. Let A_H be the DFA obtained by applying the subset construction to B_H. Note that the states of A_H are subsets of states of B_H.

Consider $a \in \Sigma$ with $(a, b) \in M$ and $\delta(q, a) = q'$ is defined for some $q' \in Q$. In other words, q has an outgoing transition on a. Assuming that (a, b) is non-trivial and $\operatorname{im} \delta_b$ contains useful states, for any set $P \subseteq Q$, we must have $\operatorname{im} \delta_b \subseteq P$ if $q \in P$. This is because for each symbol $a \in \Sigma$ for which there is a pair $(a, b) \in M$, if the NFA B_H enters a state $q \in Q$ with an outgoing transition on a, the NFA B_H also simultaneously, via ε-transitions, enters any state with an incoming transition on b. This implies that not all $2^n - 1$ non-empty subsets of Q are reachable in A_H, since the singleton set $\{q\}$ is unreachable.

Because of this construction, the number of distinct sets that contains q decreases as the size of $\operatorname{im} \delta_b$ grows. Thus, to maximize the number of sets that can be reached, the number of states with incoming transitions on any symbol b with $(a, b) \in M$ must be minimized. Therefore, for $(a, b) \in M$, there can be only one useful state with incoming transitions on b. Let us call this state $q_b \in Q$.

We claim that to maximize the number of states, A must contain no useless states and therefore A contains no sink state. First, suppose otherwise and that A contains a sink state $q_\emptyset$. To maximize the number of states, we minimize the number of states of A with outgoing transitions, so there is only one state of A, say q', with an outgoing transition on a. We observe that $q' \neq q_b$, since otherwise, $|\operatorname{im} \delta_b| = 1$ and if the only state with an outgoing transition on a is q_b itself, then the only reachable subset that contains q_b is the singleton set $\{q_b\}$.

Now, recall that for all subsets $P \subseteq Q \setminus \{q_\emptyset\}$, the two sets P and $P \cup \{q_\emptyset\}$ are indistinguishable. Then there are at most 2^{n-2} distinguishable subsets containing q_b and at most $2^{n-3} - 1$ nonempty subsets of $Q \setminus \{q_b, q', q_\emptyset\}$. Together with the sink state, this gives a total of at most $2^{n-2} + 2^{n-3}$ states in A_H.

Now, we consider when A contains no sink state. In this case, since A must be a complete DFA, in order to satisfy the condition that $|\operatorname{im} \delta_b|$ is minimal, we must have $\delta(q, a) = q_b$ for all $q \in Q$. But this means that for any state $q \in Q$ and subset $P \subseteq Q$, if $q \in P$, then $q_b \in P$. Therefore, every reachable subset of Q must contain q_b. This gives an upper bound of 2^{n-1} states in A_H.

Since $2^{n-1} > 2^{n-2} + 2^{n-3}$ for $n \geq 3$, the DFA A_H can have at most 2^{n-1} states in the worst case. □

The bound of Proposition 6 is reachable when the initial language is a regular language, even when restricted to simple splicing rules defined over an alphabet of size 3. This upper bound is met by the (2,3)-simple splicing system $H = (\Sigma, \{(c, c)\}, L(A_n))$, where $\Sigma = \{a, b, c\}$ and A_n is the DFA shown in Fig. 1. This gives us the following result.

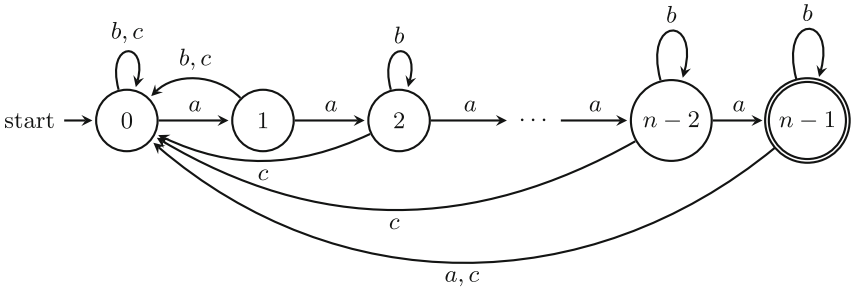

Fig. 1. The DFA A_n of Theorem 7

Theorem 7. *Let $H = (\Sigma, M, L)$ be a (2,3)-semi-simple splicing system with a regular initial language, where $L \subseteq \Sigma^*$ is a regular language with state complexity n and $M \subseteq \Sigma \times \Sigma$. Then the state complexity of $L(H)$ is at most 2^{n-1} and this bound can be reached in the worst case.*

The bound of Proposition 6 depends on whether or not the DFA for the initial language contains a sink state. Since a DFA recognizing a finite language must have a sink state, the upper bound stated in the proposition is clearly not reachable when the initial language is finite.

Proposition 8. *Let $H = (\Sigma, M, I)$ be a (2,3)-semi-simple splicing system where I is a finite language recognized by a DFA A with n states. Then a DFA recognizing $L(H)$ requires at most $2^{n-3} + 2$ states.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ be the DFA for I and let A_H be the DFA obtained via the construction of Proposition 6, given the (2,3)-semi-simple splicing system

H. We will consider the number of reachable and pairwise distinguishable states of A_H.

Recall from the proof of Proposition 6 that to maximize the number of sets that can be reached in A_H, the number of states with incoming transitions on any symbol b with $(a, b) \in M$ must be minimized. Then for $(a, b) \in M$, there can be only one useful state with incoming transitions on b. Let us call this state $q_b \in Q$.

Since I is a finite language, we know that q_0, the initial state of A, is contained in exactly one reachable state in A_H. Similarly A must contain a sink state $q_\emptyset$ and for all subsets $P \subseteq Q$, we have that P and $P \cup \{q_\emptyset\}$ are indistinguishable. Finally, we observe that there must exist at least one state $q_1 \in Q$ that is directly reachable from q_0 and is not reachable by any word of length greater than 1. Therefore, in order to maximize the number of reachable subsets, we must have that $q_1 = q_b$.

Let Q^a denote the set of states for which there is an outgoing transition on the symbol a. That is, if $q \in Q^a$, we have $\delta(q, a) \leq n - 2$. Let $k_a = |Q^a|$. It is clear that $k_a \geq 1$. Now, consider a reachable subset $P \subseteq Q \setminus \{q_0, q_\emptyset\}$. We claim that if $|P| \geq 2$ and $q_b \in P$, then we must have $q \in P$ for some $q \in Q^a$.

Suppose otherwise and that $Q^a \cap P = \emptyset$. Recall that $q_b = q_1$ and the only incoming transitions to q_1 are from the initial state q_0. Then this means that $P = \{q_1\}$ and $|P| = 1$, a contradiction. Therefore, we have $Q^a \cap P \neq \emptyset$ whenever $q_b \in P$ with $|P| \geq 2$.

Now, we can count the number of reachable subsets of $Q \setminus \{q_0, q_\emptyset\}$. There are $2^{n-3-k_a}(2^{k_a} - 1)$ non-empty subsets of size greater than 1 which contain q_b and there are $2^{n-3-k_a} - 1$ non-empty subsets which do not contain q_b. Together with the initial and sink states and the set $\{q_b\}$, we have

$$2^{n-3-k_a}(2^{k_a} - 1) + 2^{n-3-k_a} - 1 + 3.$$

Thus, the DFA A_H has at most $2^{n-3} + 2$ reachable states. □

Let $H = (\Sigma, \{(a, c)\}, L(B_n))$ be a (2,3)-semi-simple splicing system, where $\Sigma = \{a, b, c\}$ and B_n is a DFA for a finite language with n states. The DFA B_n is shown in Fig. 2. Then H is a (2,3)-semi-simple splicing system with an initial finite language that is defined over a fixed alphabet that can reach the upper bound of Proposition 8. This then gives us the following theorem.

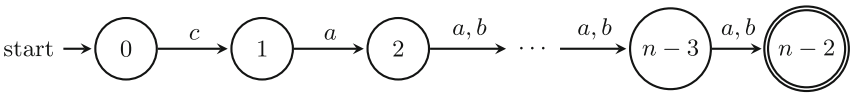

Fig. 2. The DFA B_n of Theorem 9. Transitions not shown are to the sink state $n - 1$, which is not shown.

Theorem 9. *Let $H = (\Sigma, M, I)$ be a (2,3)-semi-simple splicing system with a finite initial language, where I is a finite language with state complexity n and $M \subseteq \Sigma \times \Sigma$. Then the state complexity of $L(H)$ is at most $2^{n-3} + 2$ and this bound can be reached in the worst case.*

Unlike the situation with (2,3)-semi-simple splicing systems with regular initial languages, when we restrict (2,3)-semi-simple splicing systems with initial finite languages to allow only (2,3)-simple splicing rules, the bound of Theorem 9 is not reachable.

Proposition 10. *Let $H = (\Sigma, M, I)$ be a (2,3)-simple splicing system where I is a finite language recognized by a DFA A with n states. Then a DFA recognizing $L(H)$ requires at most $2^{n-4} + 2^{n-5} + 2$ states.*

This bound is reachable by a family of witnesses defined over an alphabet of size 7. We define the (2,3)-finite simple splicing system $H = (\Sigma, \{(c, c)\}, L(C_n))$, where $\Sigma = \{a, b, c, d, e, f, g\}$ and C_n is a DFA with n states that accepts a finite language, shown in Fig. 3. Then we have the following theorem.

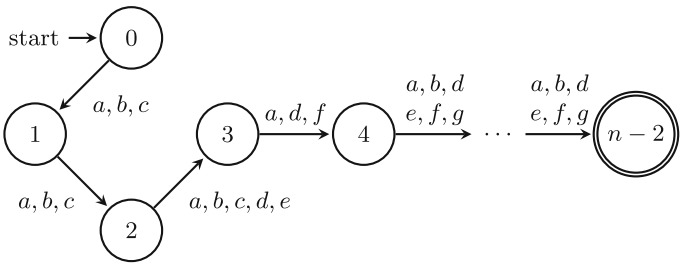

Fig. 3. The DFA C_n of Theorem 11. Transitions not shown are to the sink state $n-1$, which is not shown.

Theorem 11. *Let $H = (\Sigma, M, I)$ be a (2,3)-simple splicing system with a finite initial language, where $I \subseteq \Sigma^*$ is a finite language with state complexity n and $M \subseteq \Sigma^* \times \Sigma^*$. Then the state complexity of $L(H)$ is at most $2^{n-4} + 2^{n-5} + 2$ and this bound can be reached in the worst case.*

6 State Complexity of (1,4)-semi-simple Splicing Systems

In this section, we consider the state complexity of (1,4)-semi-simple splicing systems. Recall that a (1,4)-semi-simple splicing system is a splicing system with rules of the form $(a, \varepsilon; \varepsilon, b)$ for $a, b \in \Sigma$. As with (2,3)-semi-simple splicing systems, we can easily modify the construction of Proposition 1 to obtain an NFA for (1,4)-semi-simple splicing systems.

Proposition 12. *Let $H = (\Sigma, M, L)$ be a (1,4)-semi-simple splicing system with a regular initial language, $M = M_1 \times M_2$ with $M_1, M_2 \subseteq \Sigma$ and let L be recognized by a DFA with n states. Then there exists an NFA A'_H with $n + m$ states such that $L(A'_H) = L(H)$, where $m = |M_1|$.*

This NFA construction differs from the constructions for (2,3)- and (2,4)-semi-simple splicing systems in that additional states are introduced for each splicing rule. For each (1,4)-semi-simple splicing rule (a, b), we add a new state p_a to which any state with an outgoing transition on a has additional transitions on a and from which there are transitions on b to every state with an incoming transition on b.

This construction immediately gives an upper bound of 2^{n+m} states necessary for an equivalent DFA via the subset construction, where m is the number of symbols on the left side of each pair of rules in M. However, we will show via the following DFA construction that the upper bound is much lower than this.

Proposition 13. *Let $H = (\Sigma, M, L)$ be a (1,4)-semi-simple splicing system with a regular initial language, where $M = M_1 \times M_2$ with $M_1, M_2 \subseteq \Sigma$ and $L \subseteq \Sigma^*$ is recognized by a DFA with n states. Then there exists a DFA A_H such that $L(A_H) = L(H)$ and A_H has at most $(2^n - 2)(|M_1| + 1) + 1$ states.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. We will define the DFA $A_H = (Q', \Sigma, \delta', q_0', F')$. Then the state set of A_H is $Q' = 2^Q \times (M_1 \cup \{\varepsilon\})$, the initial state is $q_0' = \langle \{q_0\}, \varepsilon \rangle$, the set of final states is $F' = \{\langle P, a \rangle \mid P \cap F \neq \emptyset\}$, and the transition function δ' is defined

- $\delta'(\langle P, \varepsilon \rangle, a) = \langle P', \varepsilon \rangle$ if $a \notin M_1$,
- $\delta'(\langle P, \varepsilon \rangle, a) = \langle P', a \rangle$ if $a \in M_1$,
- $\delta'(\langle P, b \rangle, a) = \langle P', \varepsilon \rangle$ if $(b, a) \notin M$ and $a \notin M_1$,
- $\delta'(\langle P, b \rangle, a) = \langle P', a \rangle$ if $(b, a) \notin M$ and $a \in M_1$,
- $\delta'(\langle P, b \rangle, a) = \langle \text{ im } \delta_a, \varepsilon \rangle$ if $(b, a) \in M$ and $a \notin M_1$,
- $\delta'(\langle P, b \rangle, a) = \langle \text{ im } \delta_a, a \rangle$ if $(b, a) \in M$ and $a \in M_1$,

where $P' = \bigcup_{q \in P} \delta(q, a)$.

This construction gives an immediate upper bound of $(2^n - 1)(|M_1| + 1)$ states, however, not all of these states are distinguishable. Consider the two states $\langle Q, \varepsilon \rangle$ and $\langle Q, a \rangle$ for some $a \in M_1$. We claim that these two states are indistinguishable. This arises from the observation that $\bigcup_{q \in Q} \delta(q, a) = \text{ im } \delta_a$ for all $a \in \Sigma$. Then one of the following occurs:

- $\langle Q, \varepsilon \rangle \xrightarrow{b} \langle \text{ im } \delta_b, \varepsilon \rangle$ and $\langle Q, a \rangle \xrightarrow{b} \langle \text{ im } \delta_b, \varepsilon \rangle$ if $b \notin M_1$,
- $\langle Q, \varepsilon \rangle \xrightarrow{b} \langle \text{ im } \delta_b, b \rangle$ and $\langle Q, a \rangle \xrightarrow{b} \langle \text{ im } \delta_b, b \rangle$ if $b \in M_1$.

Note that in either case, it does not matter whether or not $(a, b) \in M$ and the two cases are distinguished solely by whether or not b is in M_1. Thus, all states $\langle Q, a \rangle$ with $a \in M_1 \cup \{\varepsilon\}$ are indistinguishable.

Thus, A_H has at most $(2^n - 2)(|M_1| + 1) + 1$ states. □

When the initial language is a regular language, the upper bound is easily reached, even when we are restricted to simple splicing rules. We consider the (1,4)-simple splicing system $H = (\Sigma, \{(c, c)\}, L(D_n))$, where $\Sigma = \{a, b, c\}$ and D_n is the DFA shown in Fig. 4.

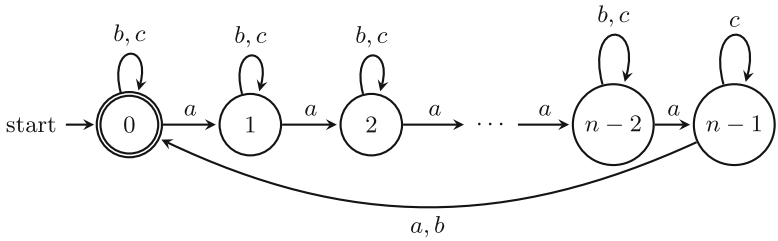

Fig. 4. The DFA D_n for Theorem 14

We note that the witness, H has $|M| = 1$ and therefore $|M_1| = 1$. We observe that we can set $|M_1|$ to be arbitrarily large by adding symbols and transitions appropriately and adding the corresponding markers to M for each new such symbol. We then have the following result.

Theorem 14. *Let $H = (\Sigma, M, L)$ be a (1,4)-semi-simple splicing system with a regular initial language, where $L \subseteq \Sigma^*$ is a regular language with state complexity n and $M = M_1 \times M_2$ with $M_1, M_2 \subseteq \Sigma$. Then the state complexity of $L(H)$ is at most $(2^n - 2)(|M_1| + 1) + 1$ and this bound can be reached in the worst case.*

We will show that this bound cannot be reached by any (1,4)-semi-simple splicing system when the initial language is finite.

Proposition 15. *Let $H = (\Sigma, M, I)$ be a (1,4)-semi-simple splicing system with a finite initial language, where $M = M_1 \times M_2$ with $M_1, M_2 \subseteq \Sigma$ and $I \subseteq \Sigma^*$ is a finite language recognized by a DFA with n states. Then there exists a DFA A_H such that $L(A_H) = L(H)$ and A_H has at most $2^{n-2} + |M_1| \cdot 2^{n-3} + 1$ states.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA for I with n states and let A_H be the DFA recognizing $L(H)$ obtained via the construction of Proposition 13. Since I is finite, the initial state of A contains no incoming transitions and A must have a sink state. Therefore, for any state $\langle S, c\rangle$, we have $S \subseteq Q \setminus \{q_0, q_\emptyset\}$ and $c \in M_1 \cup \{\varepsilon\}$, where $q_\emptyset$ is the sink state. This gives us up to $(2^{n-2}-1)(|M_1|+1)+2$ states.

We can reduce the number of reachable states further by noting that since I is finite, A must contain at least one useful state q_1 that is directly reachable only from the initial state q_0. Then there are only two ways to reach a state $\langle P, c\rangle$ in A_H with $q_1 \in P$. Either $P = \{q_1\}$ and is reached directly via a transition from $\{q_0\}$ or $|P| \geq 2$ and $P = \text{ im } \delta_b$ for some $(a, b) \in M$. For each $c \in M_1$, this gives a total of 2 reachable states $\langle P, c\rangle$.

Therefore, we can enumerate the reachable states of A_H as follows:

- the initial state $\langle \{q_0\}, \varepsilon\rangle$ and the sink state $\langle \{q_\emptyset\}, \varepsilon\rangle$,
- at most $2^{n-2} - 1$ states of the form $\langle P, \varepsilon\rangle$, where $P \subseteq Q \setminus \{q_0, q_\emptyset\}$,
- at most $|M_1|$ states of the form $\langle \{q_1\}, c\rangle$ with $c \in M_1$,
- at most $|M_1|$ states of the form $\langle P, c\rangle$ such that $P \subseteq Q \setminus \{q_0, q_\emptyset\}$, $|P| \geq 2$, and $q_1 \in P$ with $c \in M_1$,

– at most $|M_1|(2^{n-3}-1)$ states of the form $\langle P, c\rangle$ such that $P \subseteq Q \setminus \{q_0, q_1, q_\emptyset\}$ with $c \in M_1$.

This gives a total of at most $2^{n-2} + |M_1| \cdot (2^{n-3}+1) + 1$ reachable states in A_H. □

This bound is witnessed by a (1,4)-semi-simple splicing system that is defined over an alphabet and ruleset that grows exponentially in the size of the number of states of the initial language. This is similar to the (2,4)-semi-simple case. We note also that one can arbitrarily increase the size of M by adding symbols and corresponding pairs of rules appropriately. We then get the following result.

Theorem 16. *Let $H = (\Sigma, M, I)$ be a (1,4)-semi-simple splicing system with a finite initial language, where $I \subseteq \Sigma^*$ is a finite language with state complexity n and $M = M_1 \times M_2$ with $M_1, M_2 \subseteq \Sigma$. Then the state complexity of $L(H)$ is at most $2^{n-2} + |M_1| \cdot 2^{n-3} + 1$ and this bound is reachable in the worst case.*

7 Conclusion

We have studied the state complexity of several variants of semi-simple splicing systems. Our results are summarized in Table 1 and we include the state complexity of (1,3)-semi-simple and (1,3)-simple splicing systems from [9] for comparison.

Table 1. Summary of state complexity bounds for (i, j)-simple and semi-simple splicing systems with alphabet Σ, state complexity of the axiom set n, and set of splicing rules $M = M_1 \times M_2$, with $M_1, M_2 \subseteq \Sigma$. Regular axiom sets have $|\Sigma| = 3$.

	Regular axiom set	Finite axiom set	$\lvert\Sigma\rvert$
(2,4)-semi.	$2^n - 1$	$2^{n-2} + 1$	$\geq 2^{n-3}$
(2,3)-semi.	2^{n-1}	$2^{n-3} + 2$	3
(1,4)-semi.	$(2^{n-2} - 2)(\lvert M_1\rvert + 1) + 1$	$2^{n-2} + \lvert M_1\rvert \cdot 2^{n-3}$	$\geq 2^{n-3}$
(1,3)-semi. [9]	$2^n - 1$	$2^{n-2} + 1$	3
(2,4)-simple	$2^n - 1$	*Same as (1,3)*	
(2,3)-simple	2^{n-1}	$2^{n-4} + 2^{n-5} + 2$	7
(1,4)-simple	$(2^{n-2} - 2)(\lvert M_1\rvert + 1) + 1$	*?*	
(1,3)-simple [9]	$2^n - 1$	$2^{n-2} + 1$	$\geq 2^{n-3}$

Observe that for all variants of semi-simple splicing systems, the state complexity bounds for splicing systems with regular initial languages are reached with simple splicing witnesses defined over a three-letter alphabet. For semi-simple splicing systems with finite initial languages, we note that the state complexity bounds for the (2,3) and (1,3) variants are reached by witnesses defined

over a three-letter alphabet, while both of the (1,4) and (2,4) variants require an alphabet size that is exponential in the size of the DFA for the initial language.

We note that the witness for (2,3)-simple splicing systems with a finite initial language is defined over a fixed alphabet of size 7, while the problem remains open for (1,4)-simple splicing systems. Another problem that remains open is the state complexity of (1,4)- and (2,4)- simple and semi-simple splicing systems with finite initial languages defined over alphabets of size k for $3 < k < 2^{n-3}$. A similar question can be asked of (2,3)-simple splicing systems with a finite initial language for alphabets of size less than 7.

References

1. Bonizzoni, P., Ferretti, C., Mauri, G., Zizza, R.: Separating some splicing models. Inf. Process. Lett. **79**(6), 255–259 (2001)
2. Ceterchi, R., Martín-Vide, C., Subramanian, K.G.: On some classes of splicing languages. In: Aspects of Molecular Computing: Essays Dedicated to Tom Head, on the Occasion of His 70th Birthday, pp. 84–105 (2003)
3. Culik II, K., Harju, T.: Splicing semigroups of dominoes and DNA. Disc. Appl. Math. **31**(3), 261–277 (1991)
4. Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. J. Autom. Lang. Comb. **21**(4), 251–310 (2016)
5. Gatterdam, R.: Splicing systems and regularity. Int. J. Comput. Math. **31**(1–2), 63–67 (1989)
6. Goode, E., Pixton, D.: Semi-simple splicing systems. In: Martín-Vide, C., Mitrana, V. (eds.) Where Mathematics, Computer Science, Linguistics and Biology Meet, pp. 343–352. Springer, Dordrecht (2001). https://doi.org/10.1007/978-94-015-9634-3_30
7. Head, T.: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. Bull. Math. Biol. **49**(6), 737–759 (1987)
8. Head, T., Pixton, D.: Splicing and regularity. In: Recent Advances in Formal Languages and Applications, Studies in Computational Intelligence, vol. 25, pp. 119–147. Springer (2006)
9. Kari, L., Ng, T.: State complexity of simple splicing. In: Hospodár, M., Jirásková, G., Konstantinidis, S. (eds.) DCFS 2019. LNCS, vol. 11612, pp. 197–209. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23247-4_15
10. Loos, R., Malcher, A., Wotschke, D.: Descriptional complexity of splicing systems. Int. J. Found. Comput. Sci. **19**(04), 813–826 (2008)
11. Mateescu, A., Păun, G., Rozenberg, G., Salomaa, A.: Simple splicing systems. Disc. Appl. Math. **84**(1–3), 145–163 (1998)
12. Pixton, D.: Regularity of splicing languages. Disc. Appl. Math. **69**(1–2), 101–124 (1996)
13. Păun, G.: On the splicing operation. Disc. Appl. Math. **70**(1), 57–79 (1996)