

Deciding if a Regular Language is Generated by a Splicing System*

Lila Kari Steffen Kopecki

December 22, 2011

Department of Computer Science
The University of Western Ontario
{lila,steffen}@csd.uwo.ca

Abstract

Splicing as a binary word/language operation is inspired by the DNA recombination under the action of restriction enzymes and ligases, and was first introduced by Tom Head in 1987. Shortly after, it was proven that the languages generated by (finite) splicing systems form a proper subclass of the class of regular languages. However, the question of whether or not one can decide if a given regular language is generated by a splicing system remained open. In this paper we give a positive answer to this question. We namely prove that if a language is generated by a splicing system, then it is also generated by a splicing system whose size is a function of the size of the syntactic monoid of the input language, and which can be effectively constructed.

1 Introduction

In [6, 7] Head described an operation on formal languages, called *splicing*, which models DNA recombination, a cut-and-paste like operation on DNA strands under the action of restriction enzymes and ligases. A splicing system consists of a set of *axioms* or *initial words* and a set of (*splicing*) *rules*. The most commonly used definition for a rule is a quadruple 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 [2, 11].

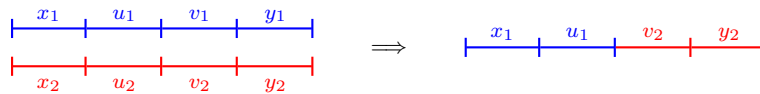


Figure 1: Splicing of the words $x_1u_1v_1y_1$ and $x_2u_2v_2y_2$ by the rule $r = (u_1, v_1; u_2, v_2)$.

Splicing as a language-theoretic word operation is meant to abstract the action of two compatible restriction enzymes and the ligase enzyme on two DNA strands. The first enzyme recognizes the subword u_1v_1 , called its *restriction site*, in any DNA string and cuts the string containing this subword between u_1 and u_2 . The second restriction enzyme, with restriction site u_2v_2 , acts similarly. Assuming that the “sticky ends” obtained after these cuts are in some sense “compatible”, the enzyme ligase aids then the recombination (catenation) of the first segment of one cut string with the second segment of another cut string.

*This research was supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant R2824A01 and Canada Research Chair Award to L. K.

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 is to consider finite sets of axioms and a finite sets of rules, which is often referred to as finite splicing systems. In this paper, by a splicing system we always mean a finite splicing system. Shortly after the introduction of splicing in formal language theory, Culik II and Harju [1] proved that splicing systems generate regular languages, only; see also [10]. However, Gatterdam [3] gave $(aa)^*$ as an example for a language which cannot be generated by a splicing system.

This led to the question, of whether or not one of the known subclasses of the regular languages corresponds to the class of languages generated by splicing systems. This does not seem to be the case. Another approach was to find an algorithm which decides whether a given regular language is generated by a splicing system. This problem has been investigated by Goode, Head, and Pixton [4, 5, 8] but it has been only partially solved: It is decidable whether a regular language is generated by a reflexive splicing system. Here reflexive means, that if $(u_1, v_1; u_2, v_2)$ is a rule, then $(u_1, v_1; u_1, v_1)$ and $(u_2, v_2; u_2, v_2)$ are rules, too. It is worth mentioning that a splicing system by the original definition in [6] is always reflexive.

In this paper we settle the problem by proving that for a given regular language, it is indeed decidable whether the language is generated by a splicing system (which is not necessarily reflexive), Corollary 4.9. More precisely, if a language L is a splicing system, then it is generated by one specific splicing system whose size is a function of the size of the syntactic monoid of L , Theorem 4.7. If m is the size of the syntactic monoid, then all axioms and all components of rules have a length in $\mathcal{O}(m^2)$. By results from [8, 10], we can construct a finite automaton for the language generated by this splicing system and compare it with a finite automaton of L . Furthermore, we prove the same result for a more general variant of splicing that has been introduced by Pixton [10], Theorem 3.7.

The paper is organized as follows. In Section 2 we lay down the notation and recall some well-known results about syntactic monoids. Section 3 and Section 4 contain the proofs for both variants of splicing. Both sections can be read independently, however, the proofs use the same idea and there are many one-to-one correspondences between the lemmas of both sections. The proof of the splicing variant by Pixton (Section 3) is less technical and should be easier to understand. Even if you are not interested in this variant of splicing, it might help to read Section 3 first.

2 Notation and Definitions

We assume the reader to be familiar with the fundamental concepts of language theory; see [9].

Let Σ be an *alphabet*, Σ^* be the set of all words over Σ , and ε denote the *empty word*. A subset L of Σ^* is a *language* over Σ . Throughout this paper, we consider languages over the fixed alphabet Σ , only. Furthermore, we consider the letters of Σ to be ordered and for words $u, v \in \Sigma^*$ we denote the (*strict*) *length-lexicographical order* by $u \leq_{\ell} v$ (resp., $u <_{\ell} v$).

For a length bound $m \in \mathbb{N}$ we let $\Sigma^{\leq m}$ denote all words whose length is at most m , i. e., $\Sigma^{\leq m} = \bigcup_{i \leq m} \Sigma^i$. Analogously, we define $\Sigma^{< m} = \bigcup_{i < m} \Sigma^i$.

Let $w \in \Sigma^*$ be a word. The length of w is denoted by $|w|$. If $w = xyz$ for some $x, y, z \in \Sigma^*$, then x , y , and z are called *prefix*, *factor*, and *suffix* of w , respectively. If a prefix or suffix of w is distinct from w , it is said to be *proper*.

2.1 Syntactic Monoids

Every language L induces an *syntactic congruence* \sim_L over words such that $u \sim_L v$ if and only if for all words x, y

$$xuy \in L \iff xvy \in L.$$

The *syntactic class* (with respect to L) of a word u is $[u]_L = \{v \mid u \sim_L v\}$. The *syntactic monoid* of L is the quotient monoid

$$M_L = \Sigma^* / \sim_L = \{[u]_L \mid u \in \Sigma^*\}.$$

It is well known, that a language L is regular if and only if its syntactic monoid M_L is finite. We will use two basic facts about syntactic monoids of regular languages.

Lemma 2.1. *Let L be a regular language and let w be a word with $|w| \geq |M_L|^2$. We can factorize $w = \alpha\beta\gamma$ with $\beta \neq \varepsilon$ such that $\alpha \sim_L \alpha\beta$ and $\gamma \sim_L \beta\gamma$.*

Lemma 2.2. *Let L be a regular language. Every element $X \in M_L$ contains a word $x \in X$ with $|x| < |M_L|$.*

3 Pixton's Variant of Splicing

In this section we use the definition of the splicing operation as it was introduced in [10]. A triple of words $r = (u_1, u_2; v) \in (\Sigma^*)^3$ is called a (*splicing*) *rule*. The words u_1 and u_2 are called *left* and *right side* of r , respectively, and v is the *bridge* of r . This splicing rule can be applied to two words $w_1 = x_1u_1y_1$ and $w_2 = x_2u_2y_2$, which contain one of the sides each, in order to create the new word $z = x_1vy_2$, see Figure 2. This operation is called *splicing* and it is denoted by $(w_1, w_2) \vdash_r z$.

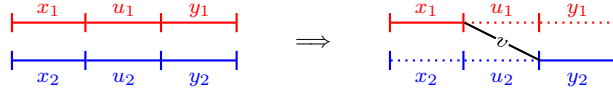


Figure 2: Splicing of the words $x_1u_1y_1$ and $x_2u_2y_2$ by the rule $r = (u_1, u_2; v)$.

It is easy to see, that the rule $(u_1v_1, u_2v_2; u_1v_2)$ expresses the same operation as the rule $(u_1, v_1; u_2, v_2)$, as defined in the introduction. However, this is a one-way relation only, which makes this variant of the splicing more general than the other one.

For a rule r we define the *splicing operator* σ_r such that for a language L

$$\sigma_r(L) = \{z \in \Sigma^* \mid \exists w_1, w_2 \in L: (w_1, w_2) \vdash_r z\}$$

and for a set of splicing rules R , we let

$$\sigma_R(L) = \bigcup_{r \in R} \sigma_r(L).$$

The reflexive and transitive closure of the splicing operator is denoted by

$$\sigma_R^*(L) = \bigcup_{i \geq 0} \sigma_R^i(L).$$

A finite set of axioms $I \subseteq \Sigma^*$ and a finite set of splicing rules $R \subseteq (\Sigma^*)^3$ form a *splicing system* (I, R) . Every splicing system (I, R) generates a language $L(I, R) = \sigma_R^*(I)$. Note that $L(I, R)$ is the smallest language which is closed under the splicing operator σ_R and includes I . It is known that the language generated by a splicing system is regular; see [10]. A (regular) language L is called a *splicing language* if a splicing system (I, R) exists such that $L = L(I, R)$.

A rule r is said to *respect* a language L if $\sigma_r(L) \subseteq L$. It is easy to see that for any splicing system (I, R) , every rule $r \in R$ respects the generated language $L(I, R)$. Moreover, a rule $r \notin R$ respects $L(I, R)$ if and only if $L(I, R \cup \{r\}) = L(I, R)$.

3.1 Rule Modifications

In this section we show, how we may modify rules that respect a language L , in order to obtain new rules which also respect L . The first lemma tells us that we may extend the sides and the bridge of a rule.

Lemma 3.1. *Let $r = (u_1, u_2; v)$ be a rule which respects a language L . For every word x , the rules $(xu_1, u_2; xv)$, $(u_1x, u_2; v)$, $(u_1, xu_2; v)$, and $(u_1, u_2x; vx)$ respect L as well.*

Proof. Let s be any of the four rules $(xu_1, u_2; xv)$, $(u_1x, u_2; v)$, $(u_1, xu_2; v)$, or $(u_1, u_2x; vx)$. In order to prove that s respects L we have to show that, for all $w_1, w_2 \in L$ and $z \in \Sigma^*$ such that $(w_1, w_2) \vdash_s z$, we have $z \in L$, too. Indeed, if $(w_1, w_2) \vdash_s z$, then $(w_1, w_2) \vdash_r z$ and as r respects L , we conclude $z \in L$. \square

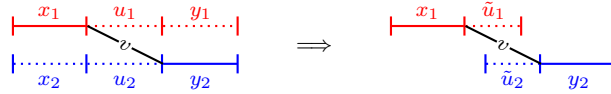
Henceforth, we will refer to the rules $(xu_1, u_2; xv)$ and $(u_1, u_2x; vx)$ as extension of the bridge and to the rules $(u_1x, u_2; v)$ and $(u_1, xu_2; v)$ as extensions of the left and right side, respectively.

Next, for a language L , let us investigate the syntactic class of a rule $r = (u_1, u_2; v)$. The *syntactic class* (with respect to L) of r is the set of rules $[r]_L = [u_1]_L \times [u_2]_L \times [v]_L$ and two rules r and s are *syntactically congruent* (with respect to L), denoted by $r \sim_L s$, if $s \in [r]_L$.

Lemma 3.2. *Let r be a rule which respects a language L . Every rule $s \in [r]_L$ respects L .*

Proof. Let $r = (u_1, u_2; v)$ and $s = (\tilde{u}_1, \tilde{u}_2; \tilde{v})$. Thus, $u_i \sim_L \tilde{u}_i$ for $i = 1, 2$ and $v \sim_L \tilde{v}$. For $\tilde{w}_1 = x_1\tilde{u}_1y_1 \in L$ and $\tilde{w}_2 = x_2\tilde{u}_2y_2 \in L$ we have to show that $\tilde{z} = x_1\tilde{v}y_2 \in L$. For $i = 1, 2$, let $w_i = x_iu_iy_i$ and note that $w_i \sim_L \tilde{w}_i$; hence, $w_i \in L$. Furthermore, $(w_1, w_2) \vdash_r x_1vy_2 = z \in L$ as r respects L and $\tilde{z} \in L$ as $z \sim_L \tilde{z}$. \square

By using Lemmas 3.1 and 3.2 we can establish length bounds for the sides of rules in case when L is a regular language. We also get rid of the factors y_1 and x_2 which do not contribute to the splicing at all, just as in the following figure.

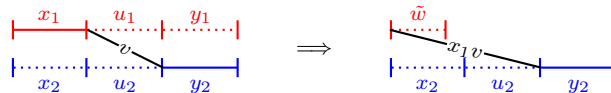


Lemma 3.3. *Let $r = (u_1, u_2; v)$ be a rule which respects a regular language L and $w_1 = x_1u_1y_1 \in L$, $w_2 = x_2u_2y_2 \in L$. There is a rule $s = (\tilde{u}_1, \tilde{u}_2; v)$ which respects L and words $\tilde{w}_1 = x_1\tilde{u}_1 \in L$, $\tilde{w}_2 = \tilde{u}_2y_2 \in L$ such that $|\tilde{u}_1|, |\tilde{u}_2| < |M_L|$. More precisely, $\tilde{u}_1 \in [u_1y_1]_L$ and $\tilde{u}_2 \in [x_2u_2]_L$.*

In particular, whenever $(w_1, w_2) \vdash_r x_1vy_2 = z$, then there are words \tilde{w}_1, \tilde{w}_2 and a rule s , as above, such that $(\tilde{w}_1, \tilde{w}_2) \vdash_s z$.

Proof. By Lemma 3.1, the rule $(u_1y_1, x_2u_2; v)$ respects L . Choose $\tilde{u}_1 \in [u_1y_1]_L$ and $\tilde{u}_2 \in [x_2u_2]_L$ as shortest words from the syntactic classes, respectively; as such, $|\tilde{u}_1|, |\tilde{u}_2| < |M_L|$ and $\tilde{w}_1 = x_1\tilde{u}_1 \in L$, $w_2 = \tilde{u}_2y_2 \in L$. Furthermore, by Lemma 3.2, $s = (\tilde{u}_1, \tilde{u}_2; v)$ respects L . \square

The next lemma allows us to modify the left side of a rule, by extending the bridge of the rule until it becomes a prefix of w_1 . As the splicing operation is perfectly symmetric, we can modify the right side of the rule, accordingly, even though Lemma 3.4 does not explicitly mention this.

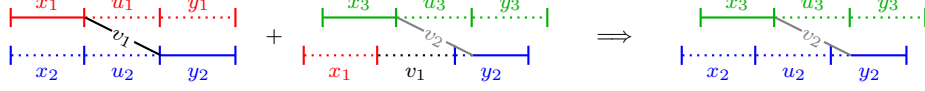


Lemma 3.4. *Let $r = (u_1, u_2; v)$ be a rule which respects a regular language L and let $w = x_1u_1y_1 \in L$. Every rule $s = (\tilde{w}, u_2; x_1v)$ where $\tilde{w} \in [w]_L \subseteq L$ respects L . In particular, there is a rule s , as above, where $|\tilde{w}| < |M_L|$.*

Proof. By Lemma 3.1, we see that $(x_1u_1y_1, u_2; x_1v)$ respects L and, by Lemma 3.2, $s = (\tilde{w}, u_2; x_1v)$ respects L . If $\tilde{w} \in [w]_L$ is a shortest word from the set, then $|\tilde{w}| < |M_L|$. \square

3.2 Series of Splicings

Consider a word z which is created by two successive splicings from words w_1, w_2 , and w_3 , as in the following figure. If no factor of w_1 or of the bridge in the first splicing is a part of z , then we can find another splicing rule s such that $(w_3, w_2) \vdash_s z$ and the bridge of s is the bridge used in the second splicing.



Lemma 3.5. *Let L be a language, $w_i = x_i u_i y_i \in L$ for $i = 1, 2, 3$, and $r_1 = (u_1, u_2; v_1)$, $r_2 = (u_3, u_4; v_2)$ be rules respecting L . If there are splicings*

$$(w_1, w_2) \vdash_{r_1} x_1 v_1 y_2 = w_4 = x_4 u_4 y_4, \quad (w_3, w_4) \vdash_{r_2} x_3 v_2 y_4 = z$$

where y_4 is a suffix of y_2 , then there is a rule $s = (u_3, \tilde{u}_2; v_2)$ which respects L and $(w_3, w_2) \vdash_s z$.

Proof. By extending the bridge v_1 of r_1 and the right side u_4 of r_2 (Lemma 3.1), we may assume the factors v_1 and u_4 match in w_4 . In order to do so, we may have to extend v_1 to both sides (thus, u_1 and u_2 may be modified) and we may have to extend u_4 to the left only, as y_4 is a suffix of y_2 . The left side u_3 and the bridge v_2 of r_2 are not modified. Now, we have $v_2 = u_4$, $x_1 = x_4$, and $y_2 = y_4$. Let $s = (u_3, u_2; v_2)$ (where u_2 is the modified right side of r_1). As desired, we have $(w_3, w_2) \vdash_s x_3 v_2 y_4 = z$ since $w_2 = x_2 u_2 y_4$.

Let $\tilde{w}_i = \tilde{x}_i u_i \tilde{y}_i \in L$ for $i = 2, 3$. If for all those words $\tilde{x}_3 v_2 \tilde{y}_2 \in L$, then s respects L . Indeed, we may splice

$$(w_1, \tilde{w}_2) \vdash_{r_1} x_1 v_1 \tilde{y}_2 = x_1 u_4 \tilde{y}_2, \quad (\tilde{w}_3, x_1 u_4 \tilde{y}_2) \vdash_{r_2} \tilde{x}_3 v_2 \tilde{y}_2.$$

Therefore, $\tilde{x}_3 v_2 \tilde{y}_2 \in L$ and s respects L . □

Consider a splicing system (I, R) and the generated language $L = L(I, R)$. Let n be the length of the longest word in I and let μ be the length-lexicographically largest word that is a component of a rule in R . Define $W_\mu = \{w \in \Sigma^* \mid w \leq_{\ell\ell} \mu\}$ as the set of all words that are length-lexicographically at most as large as μ . Furthermore, let $J = \Sigma^{\leq n} \cap L$ be a set of axioms and let

$$S = \{r \in W_\mu^3 \mid r \text{ respects } L\}$$

be a set of rules. It is easy to see that $I \subseteq J$, $R \subseteq S$, and $L = L(J, S)$. Whenever convenient, we may assume that a splicing language L is generated by a splicing system which is of the form of (J, S) .

Now, consider the creation of a word $xzy \in L$ by splicing in (J, S) . The creation of xzy can be traced back to a word $x_1 z y_1$ where either $x_1 z y_1 \in J$ or where $x_1 z y_1$ is created by a splicing that affects z , i. e., the bridge in this splicing overlaps with the factor z in $x_1 z y_1$. Whenever this is the case, we may modify the words and rules, that we used for creating xzy such that they satisfy certain restrictions.

Lemma 3.6. *Let L be a splicing language, let $\ell, n \in \mathbb{N}$, let $m = |M_L|$, and let μ be a word with $|\mu| \geq \ell + 2m$ such that for $I = \Sigma^{\leq n} \cap L$ and $R = \{r \in W_\mu^3 \mid r \text{ respects } L\}$ we have $L = L(I, R)$.*

Let $z_{k+1} = x_{k+1} z y_{k+1}$, with $|x_{k+1}|, |y_{k+1}| \leq \ell$, be a word that is created by k splicings from a word $z_1 = x_1 z y_1$ where either $x_1 z y_1 \in I$ or $x_1 z y_1$ is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ with $\tilde{w}_1, \tilde{w}_2 \in L$, $s \in R$, and the bridge of s overlaps with z . For $i = 1, \dots, k$ the intermediate splicings are either

- (i) $(w_i, z_i) \vdash_{r_i} x_{i+1} z y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R$, $y_{i+1} = y_i$, and the bridge of r_i is fully covered by the prefix x_{i+1} or

(ii) $(z_i, w_i) \vdash_{r_i} x_{i+1} z y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R$, $x_{i+1} = x_i$, and the bridge of r_i is fully covered by the suffix y_{i+1} .

There are rules and words creating z_{k+1} , as above, satisfying in addition:

1. There is $k' \leq k$ such that for $i = 1, \dots, k'$ all splittings are of form (i) and for $i = k' + 1, \dots, k$ all splittings are of form (ii).
2. For $i = 1, \dots, k$ the following bounds apply: $|x_i|, |y_i| < \ell + 2m$, $|w_i| < m$, $r_i \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<\ell+m}$.

In particular, if $n \geq m$, then $w_1, \dots, w_k \in I$.

Proof. The first statement follows by the simple observation that the order of two successive splittings may be changed when they do not interfere with each other, meaning that the factors created by the bridges do not overlap with each other. To be more formal, consider rules $r = (u_1, u_2; v_1)$ and $s = (\tilde{u}_2, u_3; v_2)$ and words $w_1 = x_1 u_1 y_1$, $w_2 = x_2 u_2 \tilde{w} \tilde{u}_2 y_2$, and $w_3 = x_3 u_3 y_3$ (these notations are not supposed to match with the notations in the claim). The word $z = x_1 v_1 \tilde{w} v_2 y_3$ can be obtained by the splittings

$$\begin{aligned} (w_1, w_2) \vdash_r x_1 v_1 \tilde{w} \tilde{u}_2 y_2 = \tilde{z}, & & (\tilde{z}, w_3) \vdash_s z & \text{ as well as} \\ (w_2, w_3) \vdash_s x_2 u_2 \tilde{w} v_2 y_3 = \hat{z}, & & (w_1, \hat{z}) \vdash_r z, \end{aligned}$$

which makes the order of the splittings irrelevant.

Note that if $k = 0$, then statement 2 is trivially true. By the first statement, $x_{k'+1} = x_{k'+2} = \dots = x_{k+1}$ and $y_1 = y_2 = \dots = y_{k'+1}$. Let us consider the splittings of form (i) which are the steps $i = 1, \dots, k'$. The notation we employ in order to prove the second statement for $i = 1, \dots, k'$ are chosen to match the notation in Figure 3.

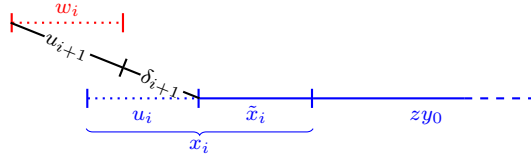


Figure 3: The i -th splitting step where $v_i = u_{i+1} \delta_{i+1}$ and $x_{i+1} = u_{i+1} \delta_{i+1} \tilde{x}_i$.

Let $r_i = (w_i, u_i; v_i)$ where $w_i \in \Sigma^{<m} \cap L$ (by Lemma 3.4) and $x_i = u_i \tilde{x}_i$ (by Lemma 3.1, we extended the side u_i to cover a prefix of x_i) such that $u_{i+1} \tilde{x}_{i+1} = v_i \tilde{x}_i$ with $u_{k'+1} = \varepsilon$ and $\tilde{x}_{k'+1} = x_{k'+1} = x_{k+1}$. Lemma 3.5 justifies the assumption that every splitting occurs to the left of the preceding splitting, i. e., \tilde{x}_i is a proper suffix of \tilde{x}_{i+1} . Note that, as $|\tilde{x}_{k'+1}| \leq \ell$, the length of \tilde{x}_i is bounded by ℓ . Now, choose δ_{i+1} such that $\tilde{x}_{i+1} = \delta_{i+1} \tilde{x}_i$; thus, $u_{i+1} \delta_{i+1} = v_i$.

For $i = 2, \dots, k'$ we replace u_i by a shortest word from $[u_i]_L$. Note that this does not change the fact, that all rules respect L (Lemma 3.2). We also replace the prefix of x_i and v_{i-1} by this factor. (There is no need to change $v_{k'}$ as $v_{k'} = \delta_{k'+1}$.) Therefore, $|x_i| < |\tilde{x}_i| + m < \ell + m$ and $r_i \in \Sigma^{<m} \times \Sigma^{<m} \times \Sigma^{<\ell+m}$ (if $i \neq 1$). We do not change u_1 yet as this may effect the splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ if it exists. Note that, for $i = 2, \dots, k'$, we have actually have proven a stronger bound than claimed in statement 2. Even though we have not proven the bound for r_1 yet, we have already established $r_1 \in \Sigma^{<m} \times \Sigma^* \times \Sigma^{<\ell+m}$. Symmetrically, we can consider statement 2 to be proven for $i = k' + 2, \dots, k$, i. e., only the prefix x_1 and the suffix $y_1 = y_{k'+1}$ have not been modified yet.

Now, let $x_1 = u_1 \tilde{x}_1$ (as above) and, symmetrically, let $y_1 = \tilde{y}_{k'+1} u_{k'+1}$ where $u_{k'+1}$ is the right side of $r_{k'+1}$. If $k' = 0$ (or $k' = k$), then u_1 (resp., $u_{k'+1}$) can be considered empty and $\tilde{x}_1 = x_{k+1}$ (resp., $\tilde{y}_{k'+1} = y_{k+1}$). If $z_1 \in I$ we replace u_1 and $u_{k'+1}$ by shortest words from their

syntactic classes, respectively, and the claim holds. Otherwise, $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ where $s = (\tilde{u}_1, \tilde{u}_2, v)$, $\tilde{w}_1 = x\tilde{u}_1$, and $\tilde{w}_2 = \tilde{u}_2y$, by Lemma 3.3. Thus,

$$z_1 = u_1\tilde{x}_1z\tilde{y}_{k'+1}u_{k'+1} = xvy$$

where $u_1\tilde{x}_1$ is a proper prefix of xv and $\tilde{y}_{k'+1}u_{k'+1}$ is a proper suffix of vy .

In case when v does not overlap with the prefix u_1 of z_1 , replace u_1 by a shortest word from its syntactic class. If v and the prefix u_1 overlap, let $u_1 = \delta_1\delta_2$ such that δ_2 is the overlap and replace δ_1 and δ_2 by a shortest word from their syntactic classes, respectively. No matter which case applied, $|u_1| < 2m$ and if v was modified, it got shorter; hence, we still have $v \in W_\mu$. Observe that $|x_1| < \ell + 2m$ and $r_1 \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<\ell+m}$. Symmetrically, we may treat $u_{k'+1}$ and $r_{k'+1}$ in order to prove statement 2. \square

3.3 Main Result

The main result of this section is:

Theorem 3.7. *Let L be a splicing language and $m = |M_L|$. The splicing system (I, R) with $I = \Sigma^{<m^2+6m} \cap L$ and*

$$R = \left\{ r \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<m^2+10m} \mid r \text{ respects } L \right\}$$

generates the language $L = L(I, R)$.

The proof is split up in two parts. In the first part, Lemma 3.8, we prove that the set of axioms can be chosen as $\Sigma^{<m^2+6m} \cap L$ and that the sides of all rules can be shorter than $2m$, but we do not care much about the lengths of the bridges.

The second part will then conclude the proof of Theorem 3.7 by showing that there are no rules with *long bridges* in R that cannot be replaced by rules with shorter bridges.

Throughout this section, by \sim we denote the equivalence relation \sim_L and by $[\cdot]$ we denote the corresponding equivalence classes $[\cdot]_L$.

Lemma 3.8. *Let I , L , and m as in Theorem 3.7. There is $n \in \mathbb{N}$ such that the splicing system (I, R') with*

$$R' = \left\{ r \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{\leq n} \mid r \text{ respects } L \right\}$$

generates the language $L = L(I, R')$.

Proof. As $I \subseteq L$ and every rule in R' respects L , it is plain that $L(I, R') \subseteq L$ (for any n); we only need to prove the converse inclusion.

As L is a splicing language, $L = L(J, S)$ for some splicing system (J, S) . Let n be larger than every bridge of every rule in S and $n \geq 4m^2$.

In order to prove $L \subseteq L(I, R')$ we use induction on the length of words in L . For all $w \in L$ with $|w| < m^2 + 6m$, by definition, $w \in I \subseteq L(I, R')$.

Now, consider $w \in L$ with $|w| \geq m^2 + 6m$. The induction hypothesis states that every word $\tilde{w} \in L$ with $|\tilde{w}| < |w|$ belongs to $L(I, R')$. Factorize $w = x\alpha\beta\gamma\delta y$ such that $|x|, |y| = 3m$, $|\alpha\beta\gamma| = m^2$, $|\beta| \geq 1$, $\alpha \sim \alpha\beta$, and $\gamma \sim \beta\gamma$.

Choose j huge ($j > n$ and J does not contain words of length j or more). We let $z = \alpha\beta^j\gamma\delta$ and investigate the creation of $xzy \in L$ by splicing in (J, S) . As z is not a factor of a word in J , we can trace back the creation of xzy by splicing to the point where the factor z is affected for the last time. Let $z_{k+1} = x_{k+1}z y_{k+1}$, where $x_{k+1} = x$ and $y_{k+1} = y$, be created by k splittings from a word $z_1 = x_1z y_1$ where $x_1z y_1$ is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ with $\tilde{w}_1, \tilde{w}_2 \in L$, $s \in S$, and the bridge of s overlaps with z . Furthermore, for $i = 1, \dots, k$ the intermediate splittings are either

- (i) $(w_i, z_i) \vdash_{r_i} x_{i+1}z y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in S$, $y_{i+1} = y_i$, and the bridge of r_i is fully covered by the prefix x_{i+1} or

(ii) $(z_i, w_i) \vdash_{r_i} x_{i+1} z y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in S$, $x_{i+1} = x_i$, and the bridge of r_i is fully covered by the suffix y_{i+1} .

Following Lemma 3.6, we may assume that $w_1, \dots, w_k \in I$, $r_1, \dots, r_k \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<4m}$, thus $r_1, \dots, r_k \in R'$, and $|x_1|, |y_1| < 5m$. Furthermore, we may use the same words and rules in order to create $w = x_{k+1} \alpha \beta \gamma \delta y_{k+1}$ from $x_1 \alpha \beta \gamma \delta y_1$ by splicing, i. e., if $x_1 \alpha \beta \gamma \delta y_1$ belongs to $L(I, R')$, so does w .

Now, consider the first splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1 = x_1 z y_1$. By Lemma 3.3, we assume $s = (u_1, u_2; v)$ such that $\tilde{w}_1 = x u_1$, $\tilde{w}_2 = u_2 y$ and $|u_1|, |u_2| < m$. (x and y are newly chosen words). Hence,

$$z_0 = x v y = x_1 \alpha \beta^j \gamma \delta y_1.$$

where x is a proper prefix of $x_1 \alpha \beta^j \gamma \delta$ and y is a proper suffix of $\alpha \beta^j \gamma \delta y_1$.

We will now pump down the factor β^j to β in order to obtain the words \hat{x} , \hat{v} , \hat{y} from x , v , y , respectively, as follows: If v overlaps with β^j but does neither cover α nor γ , extend v (Lemma 3.1) such that $v = \alpha \beta^j \gamma$. Thus, the factor $\alpha \beta^j \gamma$ is fully covered by either xv or vy . If $\alpha \beta^j$ or $\beta^j \gamma$ is fully covered by one of x , v , or y , then replace this factor by $\alpha \beta$ or $\beta \gamma$, respectively. Otherwise, by symmetry, assume that $\alpha \beta^j \gamma$ is covered by xv and, therefore, we can factorize

$$x = x_1 \alpha \beta^{j_1} \beta_1 \quad v = \beta_2 \beta^{j_2} \gamma \tilde{v}$$

where $\beta_1 \beta_2 = \beta$ and $j_1 + j_2 + 1 = j$. The results of pumping are the words $\hat{x} = x_1 \alpha \beta_1$, $\hat{v} = \beta_2 \gamma \tilde{v}$, and $\hat{y} = \tilde{y}_2$.

Let \hat{u}_1 and \hat{u}_2 be the sides of s that may have been altered due to extension and, by Lemma 3.3, assume $|\hat{u}_1|, |\hat{u}_2| < m$. If we used extension for v , then $|\hat{v}| = m^2$. No matter whether we used extension, $t = (\hat{u}_1, \hat{u}_2; \hat{v}) \in R'$ and $(\hat{x} \hat{u}_1, \hat{u}_2 \hat{y}) \vdash_t x_1 \alpha \beta \gamma \delta y_1$ as desired. Observe that \hat{x} is a prefix of $x_1 \alpha \beta \gamma \delta$ and \hat{y} is a suffix of $\alpha \beta \gamma \delta y_1$ and recall that $|x_1|, |y_1| < 5m$. Therefore, $|\hat{x} \hat{u}_1|, |\hat{u}_2 \hat{y}| < |\alpha \beta \gamma \delta| + 6m = |w|$ and, by induction hypothesis, $\hat{x} \hat{u}_1$ and $\hat{u}_2 \hat{y}$ belong to $L(I, R')$. We conclude that $x_1 \alpha \beta \gamma \delta y_1$ as well as w belong to $L(I, R')$. \square

We are now prepared to prove the main result.

Proof of Theorem 3.7. Recall that for a splicing language L with $m = |M_L|$, we intend to prove that the splicing system (I, R) with $I = \Sigma^{<m^2+6m} \cap L$ and

$$R = \left\{ r \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<m^2+10m} \mid r \text{ respects } L \right\}$$

generates the language $L = L(I, R)$. Obviously, $L(I, R) \subseteq L$. By Lemma 3.8, there is a finite set of rules $R' \subseteq \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^*$ such that $L(I, R') = L$.

For a word μ , let $W_\mu = \{w \in \Sigma^* \mid w \leq_{\ell\ell} \mu\}$, as before, and define the finite set of rules

$$R_\mu = \left\{ r \in \Sigma^{<2m} \times \Sigma^{<2m} \times W_\mu \mid r \text{ respects } L \right\}$$

and the language $L_\mu = L(I, R_\mu) \subseteq L$. If $L_\mu = L$ for some word μ , then for all v with $\mu \leq_{\ell\ell} v$, $L_v = L$ and there exists a word μ such that $R' \subseteq R_\mu$ and $L_\mu = L$. Furthermore, for $\nu = b^{m^2+10m-1}$, where b is the lexicographically largest letter in Σ , $R_\nu = R$ and, therefore, if $L_\nu = L$, the claim holds. For the sake of contradiction assume $L_\nu \neq L$ and let μ be a length-lexicographically smallest word such that $L_\mu = L$, i. e., $|\mu| \geq m^2 + 10m$. Let μ' be the length-lexicographically next-smaller word than μ and let $S = R_{\mu'}$; thus, $L(I, S) \subsetneq L$. Note that $R_\mu \setminus S$ only contains rules whose bridges are μ .

Choose w from $L \setminus L(I, S)$ as a shortest word, i. e., for all $\tilde{w} \in L$ with $|\tilde{w}| < |w|$, we have $\tilde{w} \in L(I, S)$. Factorize $w = x z y$ with $|x| = |y| = 3m$ (n. b., $|w| \geq m^2 + 6m$, otherwise $w \in I$) and factorize $\mu = \delta_1 \alpha \beta \gamma \delta_2$ with $|\delta_1|, |\delta_2| \geq 5m$, $|\alpha \beta \gamma| = m^2$, $\beta \neq \varepsilon$, $\alpha \sim \alpha \beta$, and $\gamma \sim \beta \gamma$.

Let j be a huge even number ($j > 4|\mu| + |z|$ will do). Let \tilde{z} be the word that we obtain by replacing all factors $\alpha \beta \gamma$ in z by $\alpha \beta^j \gamma$, using the following pumping algorithm:

1. let $\tilde{z} := z$;
2. if there is a factor $\alpha\beta\gamma$ of \tilde{z} such that neither
 - (a) its prefix α is succeeded by $\beta^{j/2}$ nor
 - (b) its suffix γ is preceded by $\beta^{j/2}$,
then replace this factor by $\alpha\beta^j\gamma$;
3. repeat step 2 until there is no such factor $\alpha\beta\gamma$ left.

A proof that the algorithm will terminate, hence \tilde{z} is well defined, can be found in the Appendix; see Lemma A.1. The new word \tilde{z} may still contain the factor $\alpha\beta\gamma$, but if it does, then (a) or (b) holds. By induction and as $\alpha\beta\gamma \sim \alpha\beta^j\gamma$, it is easy to see that $\tilde{z} \sim z$.

Let us trace back the creation of $x\tilde{z}y \in L$ by splicing in (I, R_μ) to a word $x_1\tilde{z}y_1$ where either $x_1\tilde{z}y_1 \in I$ or where $x_1\tilde{z}y_1$ is created by a splicing that affects \tilde{z} . Let $z_{k+1} = x_{k+1}\tilde{z}y_{k+1}$, where $x_{k+1} = x$ and $y_{k+1} = y$, be created by k splittings from a word $z_1 = x_1\tilde{z}y_1$ where either $x_1\tilde{z}y_1 \in I$ or $x_1\tilde{z}y_1$ is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ with $\tilde{w}_1, \tilde{w}_2 \in L$, $s \in R_\mu$, and the bridge of s overlaps with \tilde{z} . Furthermore, for $i = 1, \dots, k$ the intermediate splittings are either

- (i) $(w_i, z_i) \vdash_{r_i} x_{i+1}\tilde{z}y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R_\mu$, $y_{i+1} = y_i$, and the bridge of r_i is fully covered by the prefix x_{i+1} or
- (ii) $(z_i, w_i) \vdash_{r_i} x_{i+1}\tilde{z}y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R_\mu$, $x_{i+1} = x_i$, and the bridge of r_i is fully covered by the suffix y_{i+1} .

Following Lemma 3.6, we may assume that $w_1, \dots, w_k \in I$, $r_1, \dots, r_k \in \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<4m}$, thus $r_1, \dots, r_k \in S$, and $|x_1|, |y_1| < 5m$. Furthermore, we may use the same words and rules in order to create $w = x_{k+1}\tilde{z}y_{k+1}$ from $x_1\tilde{z}y_1$ by splicing. As w does not belong $L(I, S)$, the word $x_1\tilde{z}y_1$ does not belong to $L(I, S)$ neither. If z_1 was in I , then $x_1\tilde{z}y_1 \in I$ as well, as it z is as most as long as \tilde{z} .

Therefore, z_1 is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ where $s = (u_1, u_2; v)$, $\tilde{w}_1 = xu_1$, and $\tilde{w}_2 = u_2y$ where $|\tilde{u}_1|, |\tilde{u}_2| < m$, by Lemma 3.3 (here, x and y are newly chosen words). We have

$$z_1 = x_1\tilde{z}y_1 = xvy$$

where x is a proper prefix of $x_1\tilde{z}$ and y is a proper suffix of $\tilde{z}y_1$. Recall that either $s \in S$ or $v = \mu$.

However, we will see next that if $v = \mu$, there is also a rule $\tilde{s} \in S$ and slightly modified words which can be used in order to create $x_1\tilde{z}y_1$ by splicing. In this case $\mu = \delta_1\alpha\beta\gamma\delta_2$ is a factor of z_1 . As $|\delta_1|, |\delta_2| \geq 5m > |x_0|, |y_0|$, the factor $\alpha\beta\gamma$ is covered by \tilde{z} and, as such, the pumping algorithm ensured that either (a) α is succeeded by $\beta^{j/2}$ or (b) γ is preceded by $\beta^{j/2}$. Due to symmetry, we only consider the former case, in which $\gamma\delta_2$ is a prefix of a word in β^+ . Let us shorten the bridge v such that $\tilde{s} = (u_1, u_2; \delta_1\alpha\gamma\delta_2)$. Note that $\tilde{s} \in S$ (as $\alpha \sim \alpha\beta$ and by Lemma 3.2). Furthermore, as j is large enough, $y = \beta_2\beta^\ell\tilde{y}$ where β_2 is the suffix of β such that $\gamma\delta_2\beta_2 \in \beta^+$ and $\ell \geq |\gamma|$. Note that this implies $\beta_2\gamma$ is a prefix of y , which allows us to add an additional β . Therefore, $(\tilde{w}_1, u_2\beta_2\beta^{\ell+1}\tilde{y}) \vdash_{\tilde{s}} z_1$ where $u_2\beta_2\beta^{\ell+1}\tilde{y} \in L$. This observation justifies the assumption that $v \neq \mu$ and $s \in S$ which we will make for the remain of the proof.

Next, we will pump down the factors $\alpha\beta^j\gamma$ to $\alpha\beta\gamma$ in \tilde{z} again. At every position where we pumped up before, we are now pumping down (in reverse order) in order to obtain the words \hat{x} , \hat{v} , \hat{y} from the words x , v , y , respectively. The pumping in each step is done as in the proof of Lemma 3.8:

If v overlaps with β^j but does neither cover α nor γ , extend v (Lemma 3.1) such that $v = \alpha\beta^j\gamma$. Thus, the factor $\alpha\beta^j\gamma$ is fully covered by either xv or vy . If $\alpha\beta^j$ or $\beta^j\gamma$ is fully covered by one of x , v , or y , then replace this factor by $\alpha\beta$ or $\beta\gamma$, respectively. Otherwise, by symmetry, assume that $\alpha\beta^j\gamma$ is covered by xv and, therefore, we can factorize

$$x = \tilde{x}\alpha\beta^j\beta_1 \qquad v = \beta_2\beta^{j^2}\gamma\tilde{v}$$

where $\beta_1\beta_2 = \beta$ and $j_1+j_2+1 = j$. The pumping result are the words $\tilde{x}\alpha\beta_1$ and $\beta_2\gamma\tilde{v}$, respectively.

Let \hat{u}_1 and \hat{u}_2 be the sides of s that may have been altered due to extension and, by Lemma 3.3, assume $|\hat{u}_1|, |\hat{u}_2| < m$. If we used extension for v in one of the steps, then $|\hat{v}| \leq m^2$. No matter whether we used extension, $t = (\hat{u}_1, \hat{u}_2; \hat{v}) \in S$ and $(\hat{x}\hat{u}_1, \hat{u}_2\hat{y}) \vdash_t x_1zy_1$. As $|\hat{x}\hat{u}_1|, |\hat{u}_2\hat{y}| < |z| + 6m = |w|$, $\hat{x}\hat{u}_1$ and $\hat{u}_2\hat{y}$ belong to $L(I, S)$. Concluding that x_1zy_1 as well as w belong to $L(I, S)$ — the desired contradiction. \square

4 Splicing

In this section, we consider the splicing operation as defined in [2, 11]. This is most commonly used definition for splicing in formal language theory. Throughout this section, a quadruple of words $r = (u_1, v_1; u_2, v_2) \in (\Sigma^*)^4$ is called a (*splicing*) *rule*. The words u_1v_1 and u_2v_2 are called *left* and *right side* of r . This splicing rule can be applied to two words $w_1 = x_1u_1v_1y_1$ and $w_2 = x_2u_2v_2y_2$, which contain one of the sides each, in order to create the new word $z = x_1u_1v_2y_2$, see Figure 4. This operation is called *splicing* and it is denoted by $(w_1, w_2) \vdash_r z$. The *splicing position* of this splicing is the position between the factors x_1u_1 and v_2y_2 in z .

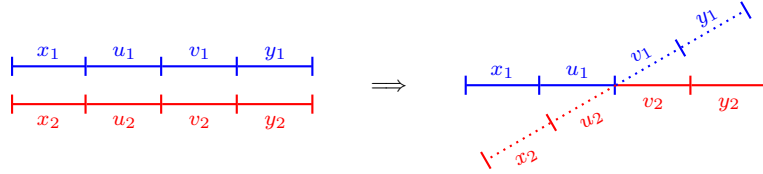


Figure 4: Splicing of the words $x_1u_1v_1y_1$ and $x_2u_2v_2y_2$ by the rule $r = (u_1, v_1; u_2, v_2)$.

Just as in Section 3, for a rule r we define the *splicing operator* σ_r such that for a language L

$$\sigma_r(L) = \{z \in \Sigma^* \mid \exists w_1, w_2 \in L: (w_1, w_2) \vdash_r z\}$$

and for a set of splicing rules R , we let

$$\sigma_R(L) = \bigcup_{r \in R} \sigma_r(L).$$

The reflexive and transitive closure of the splicing operator is denoted by

$$\sigma_R^*(L) = \bigcup_{i \geq 0} \sigma_R^i(L).$$

A finite set of axioms $I \subseteq \Sigma^*$ and a finite set of splicing rules $R \subseteq (\Sigma^*)^4$ form a *splicing system* (I, R) . Every splicing system (I, R) generates a language $L(I, R) = \sigma_R^*(I)$. Note that $L(I, R)$ is the smallest language which is closed under the splicing operator σ_R and includes I . It is known that the language generated by a splicing system is regular; see [1, 10]. A (regular) language L is called a *splicing language* if a splicing system (I, R) exists such that $L = L(I, R)$.

A rule r is said to *respect* a language L if $\sigma_r(L) \subseteq L$. It is easy to see that for any splicing system (I, R) , every rule $r \in R$ respects the generated language $L(I, R)$. Moreover, a rule $r \notin R$ respects $L(I, R)$ if and only if $L(I, R \cup \{r\}) = L(I, R)$.

4.1 Rule Modifications

Before we can prove our main result, we will make some preliminary observations. In this section we show, how we may modify rules that respect a language L , in order to obtain new rules which also respect L . The first lemma tells us, that we may extend the sides of a rule.

Lemma 4.1. *Let $r = (u_1, v_1; u_2, v_2)$ be a rule which respects a language L . For every word x , the rules $(xu_1, v_1; u_2, v_2)$, $(u_1, v_1x; u_2, v_2)$, $(u_1, v_1; xu_2, v_2)$, and $(u_1, v_1; u_2, v_2x)$ respect L as well.*

Proof. Let s be any of the rules $(xu_1, v_1; u_2, v_2)$, $(u_1, v_1x; u_2, v_2)$, $(u_1, v_1; xu_2, v_2)$, $(u_1, v_1; u_2, v_2x)$. In order to prove that s respects L we have to show that, for all $w_1, w_2 \in L$ and $z \in \Sigma^*$ such that $(w_1, w_2) \vdash_s z$, we have $z \in L$, too. Indeed, if $(w_1, w_2) \vdash_s z$, then $(w_1, w_2) \vdash_r z$ and, as r respects L , we conclude $z \in L$. \square

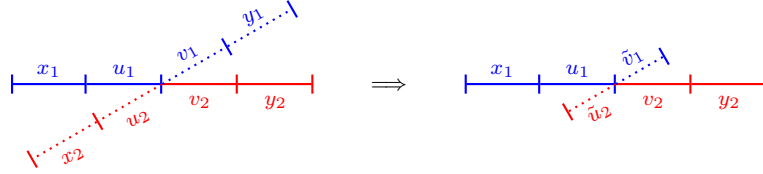
Henceforth, we will refer to the rules $(xu_1, v_1; u_2, v_2)$, $(u_1, v_1x; u_2, v_2)$ as extensions of the left side and to $(u_1, v_1; xu_2, v_2)$, $(u_1, v_1; u_2, v_2x)$ as extensions of the right side.

Next, for a language L , let us investigate the syntactic class of a rule $r = (u_1, v_1; u_2, v_2)$. The *syntactic class* (with respect to L) of r is the set of rules $[r]_L = [u_1]_L \times [v_1]_L \times [u_2]_L \times [v_2]_L$ and two rules r and s are *syntactically congruent* (with respect to L), denoted by $r \sim_L s$, if $s \in [r]_L$.

Lemma 4.2. *Let r be a rule which respects a language L . Every rule $s \in [r]_L$ respects L .*

Proof. Let $r = (u_1, v_1; u_2, v_2)$ and $s = (\tilde{u}_1, \tilde{v}_1; \tilde{u}_2, \tilde{v}_2)$. Thus, $u_i \sim_L \tilde{u}_i$ and $v_i \sim_L \tilde{v}_i$ for $i = 1, 2$. For $\tilde{w}_1 = x_1\tilde{u}_1\tilde{v}_1y_1 \in L$ and $\tilde{w}_2 = x_2\tilde{u}_2\tilde{v}_2y_2 \in L$ we have to show that $\tilde{z} = x_1\tilde{u}_1\tilde{v}_2y_2 \in L$. For $i = 1, 2$, let $w_i = x_iu_iv_iy_i$ and note that $w_i \sim_L \tilde{w}_i$; hence, $w_i \in L$. Furthermore, $(w_1, w_2) \vdash_r x_1u_1v_2y_2 = z \in L$ as r respects L , and $\tilde{z} \in L$ as $z \sim_L \tilde{z}$. \square

By using Lemmas 4.1 and 4.2 we can establish length bounds for the second and third components of rules, in case when L is a regular language. We also get rid of the factors y_1 and x_2 which do not contribute to the splicing.



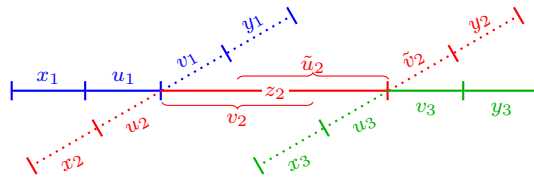
Lemma 4.3. *Let $r = (u_1, v_1; u_2, v_2)$ be a rule which respects a regular language L and $w_1 = x_1u_1v_1y_1 \in L$, $w_2 = x_2u_2v_2y_2 \in L$. There is a rule $s = (u_1, \tilde{v}_1; \tilde{u}_2, v_2)$ which respects L and words $\tilde{w}_1 = x_1u_1\tilde{v}_1 \in L$, $\tilde{w}_2 = \tilde{u}_2v_2y_2 \in L$ such that $|\tilde{v}_1|, |\tilde{u}_2| < |M_L|$. More precisely, $\tilde{v}_1 \in [v_1y_1]_L$ and $\tilde{u}_2 \in [x_2u_2]_L$.*

In particular, whenever $(w_1, w_2) \vdash_r x_1u_1v_2y_2 = z$, then there are words \tilde{w}_1, \tilde{w}_2 and a rule s , as above, such that $(\tilde{w}_1, \tilde{w}_2) \vdash_s z$.

Proof. By Lemma 4.1, the rule $(u_1, v_1y_1; x_2u_2, v_2)$ respects L . Choose $\tilde{v}_1 \in [v_1y_1]_L$ and $\tilde{u}_2 \in [x_2u_2]_L$ as shortest words from the sets, respectively; as such, $|\tilde{u}_1|, |\tilde{u}_2| < |M_L|$ and $\tilde{w}_1 = x_1u_1\tilde{v}_1 \in L$, $w_2 = \tilde{u}_2v_2y_2 \in L$. Furthermore, by Lemma 4.2, $s = (u_1, \tilde{v}_1; \tilde{u}_2, v_2)$ respects L . \square

4.2 Series of Splicings

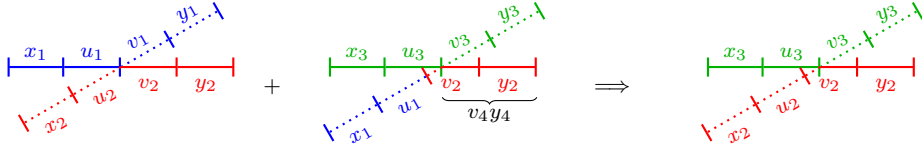
In this section, we consider the creation of words by more than one splicing. Let us begin with a simple observation. In case when a word is created by two (or more) successive splicings, but the sides of the splicing do not cover the splicing position of the other splicing, then the order of these splicings is irrelevant.



Remark 4.4. Let $w_1 = x_1u_1v_1y_1$, $w_2 = x_2u_2z_2\tilde{v}_2y_2$, where v_2 is a prefix of z_2 and \tilde{u}_2 is a suffix of z_2 , $w_3 = x_3u_3v_3y_3$ be words and $r_1 = (u_1, v_1; u_2, v_2)$, $r_2 = (\tilde{u}_2, \tilde{v}_2; u_3, v_3)$ be rules. In order to create the word $z = x_1u_1z_2v_3y_3$ by splicing, we may use splicings

$$\begin{aligned} (w_1, w_2) \vdash_{r_1} x_1u_1z_2\tilde{v}_2y_2 = z', & & (z', w_3) \vdash_{r_2} z & \text{or} \\ (w_2, w_3) \vdash_{r_2} x_2u_2z_2v_3y_3 = z'', & & (w_1, z'') \vdash_{r_1} z. & \end{aligned}$$

Consider a word z which is created by two successive splicings from words w_1 , w_2 , and w_3 , as in the following figure. If no factor of w_1 is a part of z , then we can find another splicing rule s such that $(w_3, w_2) \vdash_s z$.



Lemma 4.5. Let L be a language, $w_i = x_iu_iviy_i \in L$ for $i = 1, 2, 3$, and $r_1 = (u_1, v_1; u_2, v_2)$, $r_2 = (u_3, v_3; u_4, v_4)$ be rules respecting L . If there are splicings

$$(w_1, w_2) \vdash_{r_1} x_1u_1v_2y_2 = w_4 = x_4u_4v_4y_4, \quad (w_3, w_4) \vdash_{r_2} x_3u_3v_4y_4 = z$$

where v_4y_4 is a suffix of v_2y_2 , then there is a rule $s = (u_3, v_3; u_2\delta, \tilde{v}_4)$ which respects L and $(w_3, w_2) \vdash_s z$. Furthermore, $\tilde{v}_4 = v_4$ or $\tilde{v}_4 \leq_{\ell\ell} v_2$.

Proof. We extend u_1 , v_2 , u_4 , and v_4 such that the factors u_1v_2 and u_4v_4 match in w_4 (Lemma 4.1); more precisely, we only have to extend one of u_1, u_4 and one of v_2, v_4 . As v_4y_4 was a suffix of v_2y_2 , now, v_4 is a suffix of v_2 and either v_4 was not modified or $v_4 \leq_{\ell\ell} v_2$ and v_2 was not modified. Let δ such that $\delta v_4 = v_2$, let $s = (u_3, v_3; u_2\delta, v_4)$, and observe that $(w_3, w_2) \vdash_s z$.

Next, let us prove that s respects L . Let $\tilde{w}_3 = \tilde{x}_3u_3v_3\tilde{y}_3 \in L$ and $\tilde{w}_2 = \tilde{x}_2u_2\delta v_4\tilde{y}_2 = \tilde{x}_2u_2v_2\tilde{y}_2 \in L$. There are splicings

$$(w_1, \tilde{w}_2) \vdash_{r_1} x_1u_1v_2\tilde{y}_2 = \tilde{w}_4 = x_1u_4v_4\tilde{y}_2, \quad (\tilde{w}_3, \tilde{w}_4) \vdash_{r_2} \tilde{x}_3u_3v_4\tilde{y}_2 = \tilde{z}$$

and $\tilde{z} \in L$, concluding that s respects L . \square

Let (I, R) be a splicing system and $L = L(I, R)$, let n be the length of the longest word in I and let μ be the length-lexicographically largest word that is a component of a rule in R . Define $W_\mu = \{w \in \Sigma^* \mid w \leq_{\ell\ell} \mu\}$ as the set of all words that are length-lexicographically at most as large as μ . Furthermore, let $J = \Sigma^{\leq n} \cap L$ be a set of axioms and let

$$S = \{r \in W_\mu^4 \mid r \text{ respects } L\}$$

be a set of rules. It is not difficult to see that $I \subseteq J$, $R \subseteq S$ and $L = L(J, S)$. Whenever convenient, we will assume that a splicing language L is generated by a splicing system which is of the form of (J, S) .

We will consider a word $xzy \in L$ where the length of the middle factor z is longer than $|\mu|$. The creation of xzy by splicing in (J, S) can be traced back to a word x_1zy_1 where either $x_1zy_1 \in J$ or where x_1zy_1 is created by a splicing that affects the factor z , i. e., the splicing position lies in the factor z . Whenever this is the case, we may modify the words and rules, that we used for creating xzy such that they satisfy certain restrictions.

Lemma 4.6. Let L be a splicing language, let $\ell, n \in \mathbb{N}$, let $m = |M_L|$, and let μ be a word with $|\mu| \geq \ell + 2m$ such that for $I = \Sigma^{\leq n} \cap L$ and $R = \{r \in W_\mu^4 \mid r \text{ respects } L\}$ we have $L = L(I, R)$.

Let $z_{k+1} = x_{k+1}zy_{k+1}$ with $|z| \geq |\mu|$ and $|x_{k+1}|, |y_{k+1}| \leq \ell$ be a word that is created by k splicings from a word $z_1 = x_1zy_1$ where either $x_1zy_1 \in I$ or x_1zy_1 is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ with $\tilde{w}_1, \tilde{w}_2 \in L$, $s \in R$, and the splicing position lies in the factor z . Furthermore, for $i = 1, \dots, k$ the intermediate splicings are either

- (i) $(w_i, z_i) \vdash_{r_i} x_{i+1}zy_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R$, $y_{i+1} = y_i$, and the splicing position lies to the left of the factor z or
- (ii) $(z_i, w_i) \vdash_{r_i} x_{i+1}zy_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R$, $x_{i+1} = x_i$, and the splicing position lies to the right of the factor z .

There are rules and words creating z_{k+1} , as above, satisfying in addition:

1. There is $k' \leq k$ such that for $i = 1, \dots, k'$ all splicings are of form (i) and for $i = k'+1, \dots, k$ all splicings are of form (ii).
2. For $i = 1, \dots, k'$ the following bounds apply: $|x_i| < \ell + 2m$, $|w_i| < \ell + 2m$, $r_i \in \Sigma^{<\ell+m} \times \Sigma^{<2m} \times \Sigma^{<2m} \times W_\mu$ and $x_{k'+1} = x_{k'+2} = \dots = x_{k+1}$.
3. For $i = k'+1, \dots, k$ the following bounds apply: $|y_i| < \ell + 2m$, $|w_i| < \ell + 2m$, $r_i \in W_\mu \times \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<\ell+m}$ and $y_1 = y_2 = \dots = y_{k'+1}$.

In particular, if $n \geq \ell + 2m$, then $w_1, \dots, w_k \in I$.

Proof. The first statement follows immediately by Remark 4.4 and the fact that $|z| \geq |\mu|$. The first statement also implies $x_{k'+1} = x_{k'+2} = \dots = x_{k+1}$ and $y_1 = y_2 = \dots = y_{k'+1}$. Note that if $k' = 0$ (or $k' = k$), then statement 2 (resp., statement 3) is trivially true.

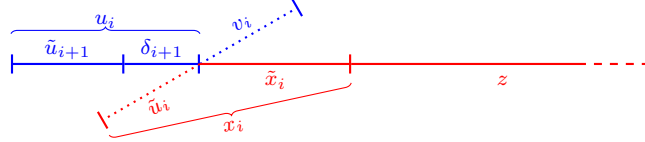


Figure 5: The i -th splicing step where $x_{i+1} = u_i \tilde{x}_i$ and \tilde{v}_i is a prefix of $\tilde{x}_i z$.

The notation we employ in order to prove statement 2 is chosen such that it matches with Figure 5. For $i = 1, \dots, k'$, let $r_i = (u_i, v_i; \tilde{u}_i, \tilde{v}_i)$. By extension (Lemma 4.1), we may assume that $w_i = u_i v_i$ and $x_i = \tilde{u}_i \tilde{x}_i$ such that $x_{i+1} = u_i \tilde{x}_i$ and \tilde{v}_i is a prefix of $\tilde{x}_i z$. Let $\tilde{x}_{k'+1} = x_{k'+1} = x_{k+1}$ and $\tilde{u}_{k'+1} = \varepsilon$. By Lemma 4.5, we may assume that every splicing position lies to the left of the previous splicing position, i. e., \tilde{x}_i is a proper suffix of \tilde{x}_{i+1} and $|\tilde{x}_i| \leq \ell$ (as $|\tilde{x}_{k+1}| \leq \ell$). Let δ_{i+1} such that $\tilde{x}_{i+1} = \delta_{i+1} \tilde{x}_i$; hence, $u_i = \tilde{u}_{i+1} \delta_{i+1}$. Now, for $i = 2, \dots, k'$, we replace \tilde{u}_i by a shortest word from $[\tilde{u}_i]_L$. (We also replace this prefix of x_i and u_{i-1} .) Furthermore, we replace v_i by a shortest word from $[v_i]_L$ for $i = 1, \dots, k'$. We do not replace \tilde{u}_1 yet, as this might affect the word \tilde{w}_1 and the rules s in the splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s x_1 z y_1$.

Observe that the words z_i , w_i , and the rules r_i can still be used to create z_{k+1} by splicing, in the way described in the claim. For $i = 2, \dots, k'$, we have $|x_i| = |\tilde{u}_i \tilde{x}_i| < \ell + m$, $|w_i| \leq |x_{i+1}| + |v_i| < \ell + 2m$, and $r_i \in \Sigma^{<\ell+m} \times \Sigma^{<m} \times \Sigma^{<m} \times W_\mu$. We also have $|w_1| < \ell + 2m$ and $r_1 \in \Sigma^{<\ell+m} \times \Sigma^{<m} \times \Sigma^* \times W_\mu$. Note that, except for the length of x_1 , and the third component of r_1 , we have proven statement 2 and we actually have proven a stronger bound than claimed. Symmetrically, we can consider statement 3 to be proven except for $y_1 = y_{k'+1}$ and the second component of $r_{k'+1}$.

Let $x_1 = \tilde{u}_1 \tilde{x}_1$ (as above) and, symmetrically, let $y_1 = \tilde{y}_{k'+1} \tilde{v}_{k'+1}$ where $\tilde{v}_{k'+1}$ is the second component of $r_{k'+1}$. If $k' = 0$ (or $k' = k$), then \tilde{u}_1 (resp., $\tilde{v}_{k'+1}$) can be considered empty and $\tilde{x}_1 = x_{k+1}$ (resp., $\tilde{y}_{k'+1} = y_{k+1}$). If $z_1 \in I$ we replace \tilde{u}_1 and $\tilde{v}_{k'+1}$ by shortest words from their syntactic classes, respectively, and the claim holds. Otherwise, $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ where x_1 is a prefix of \tilde{w}_1 and y_1 is a suffix of \tilde{w}_2 .

Let $s = (u_1, v_1; u_2, v_2)$ and consider the overlap of u_1 in this splicing with the prefix \tilde{u}_1 of \tilde{w}_1 . In case when u_1 does not overlap with \tilde{u}_1 , replace \tilde{u}_1 by a shortest word from its syntactic class. If u_1 and \tilde{u}_1 overlap, let $\tilde{u}_1 = \delta_1 \delta_2$ such that δ_2 is the overlap and replace δ_1 and δ_2 by shortest words from their syntactic classes, respectively. Note that if we modified u_1 , it got

shorter; hence, s still belongs to R . No matter which case applied, $|\tilde{u}_1| < 2m$, $|x_1| < \ell + 2m$, and $r_1 \in \Sigma^{<\ell+m} \times \Sigma^{<m} \times \Sigma^{<2m} \times W_\mu$; thus, the second statement.

We may treat $y_{k'+1}$ and $r_{k'+1}$ symmetrically in order to prove statement 3. \square

4.3 Main Result

The main result of this section is:

Theorem 4.7. *Let L be a splicing language and $m = |M_L|$. The splicing system (I, R) with $I = \Sigma^{<m^2+6m} \cap L$ and*

$$R = \left\{ r \in \Sigma^{<m^2+10m} \times \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<m^2+10m} \mid r \text{ respects } L \right\}$$

generates the language $L = L(I, R)$.

The proof is split up in two parts. In the first part, Lemma 4.8, we proof that the set of rules can be chosen as $\left\{ r \in (\Sigma^{<m^2+10m})^4 \mid r \text{ respects } L \right\}$ for some finite set of axioms.

The second part concludes the proof of Theorem 4.7, by employing the length bound $2m$ for the second and third component of rules and by proving that the set of axioms can be chosen as $\Sigma^{<m^2+6m} \cap L$.

Throughout this section, by \sim we denote the equivalence relation \sim_L and by $[\cdot]$ we denote the corresponding equivalence classes $[\cdot]_L$.

Lemma 4.8. *Let L and m as in Theorem 4.7. There exists $n \in \mathbb{N}$ such that the splicing system (I, R) with $I = \Sigma^{\leq n} \cap L$ and*

$$R = \left\{ r \in (\Sigma^{<m^2+10m})^4 \mid r \text{ respects } L \right\}$$

generates the same language $L = L(I, R)$.

Proof. As every word in I belongs to L and every rule in R respects L , the inclusion $L(I, R) \subseteq L$ holds (for any n).

Let (I', R') be a splicing system that generates $L = L(I', R')$ and let n such that $n - 6m$ is larger than any word in I' and larger than any component of rules in R' . As in the claim, let $I = \Sigma^{\leq n} \cap L$.

For a word μ we let $W_\mu = \{w \in \Sigma^* \mid w \leq_{\ell\ell} \mu\}$, as we did before. Define the set of rules where every component is length-lexicographically bounded by μ

$$R_\mu = \left\{ r \in W_\mu^4 \mid r \text{ respects } L \right\}$$

and the language $L_\mu = L(I, R_\mu)$. If $L_\mu = L$ for some word μ , then for all words v with $\mu \leq_{\ell\ell} v$, $L_v = L$ and, as $L = L(I', R')$, there exists a word μ such that $L_\mu = L$ and $|\mu| + 6m \leq n$. Furthermore, for the word $\nu = b^{m^2+10m-1}$, where b is the lexicographically largest letter in Σ , $R_\nu = R$ and, therefore, if $L_\nu = L$, the claim holds. For the sake of contradiction assume $L_\nu \neq L$ and let μ be the length-lexicographically smallest word such that $L_\mu = L$; hence, $|\mu| \geq m^2 + 10m$. Let μ' be the length-lexicographically next-smaller word than μ and let $S = R_{\mu'}$. Note that $L(I, S) \subsetneq L$ and $R_\mu \setminus S$ contains only rules which have a component that is equal to μ .

Choose w from $L \setminus L(I, S)$ as a shortest word, i.e., for all $\tilde{w} \in L$ with $|\tilde{w}| < |w|$, we have $\tilde{w} \in L(I, S)$. Factorize $w = xzy$ with $|x| = |y| = 3m$, n. b., $|z| \geq |\mu|$, otherwise $w \in I$. Factorize $\mu = \delta_1 \alpha \beta \gamma \delta_2$ with $|\delta_1|, |\delta_2| \geq 5m$, $|\alpha \beta \gamma| = m^2$, $\beta \neq \varepsilon$, $\alpha \sim \alpha \beta$, and $\gamma \sim \beta \gamma$.

Let j be a huge even number ($j > 4|\mu| + |z|$ will do). Let \tilde{z} be the word that we obtain by replacing all factors $\alpha \beta \gamma$ by $\alpha \beta^j \gamma$ in z by the following pumping algorithm:

1. let $\tilde{z} := z$;
2. if there is a factor $\alpha \beta \gamma$ of \tilde{z} such that neither

- (a) its prefix α is succeeded by $\beta^{j/2}$ nor
 - (b) its suffix γ is preceded by $\beta^{j/2}$,
- then replace this factor by $\alpha\beta^j\gamma$;

3. repeat step 2 until there is no such factor $\alpha\beta\gamma$ left.

A proof that the algorithm will terminate, hence \tilde{z} is well defined, can be found in the Appendix; see Lemma A.1. The new word \tilde{z} may still contain the factor $\alpha\beta\gamma$, but if it does, then (a) or (b) holds. By induction and as $\alpha\beta\gamma \sim \alpha\beta^j\gamma$, it is easy to see that $\tilde{z} \sim z$.

Let us trace back the creation of $x\tilde{z}y \in L$ by splicing in (I, R_μ) to a word $x_1\tilde{z}y_1$ where either $x_1\tilde{z}y_1 \in I$ or where $x_1\tilde{z}y_1$ is created by a splicing that affects \tilde{z} . Let $z_{k+1} = x_{k+1}\tilde{z}y_{k+1}$, where $x_{k+1} = x$ and $y_{k+1} = y$, be created by k splittings from a word $z_1 = x_1\tilde{z}y_1$ where either $x_1\tilde{z}y_1 \in I$ or $x_1\tilde{z}y_1$ is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ with $\tilde{w}_1, \tilde{w}_2 \in L$, $s \in R_\mu$, and the splicing position lies in the factor \tilde{z} . Furthermore, for $i = 1, \dots, k$ the intermediate splittings are either

- (i) $(w_i, z_i) \vdash_{r_i} x_{i+1}\tilde{z}y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R_\mu$, $y_{i+1} = y_i$, and the splicing position lies to the left of the factor \tilde{z} or
- (ii) $(z_i, w_i) \vdash_{r_i} x_{i+1}\tilde{z}y_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in R_\mu$, $x_{i+1} = x_i$, and the splicing position lies to the right of the factor \tilde{z} .

Note that $|\tilde{z}| \geq |z| \geq |\mu|$ and, therefore, we can apply Lemma 4.6. Thus, $w_1, \dots, w_k \in I$ and $|x_i|, |y_i| < 5m$ for $i = 1, \dots, k$.

Consider a rule r_i in a splicing of form (i). By Lemma 4.6, $r_i \in \Sigma^{<4m} \times \Sigma^{<2m} \times \Sigma^{<2m} \times W_\mu$. Suppose the forth component of r_i covers a prefix of the left-most factor $\alpha\beta^{j/2}$ in \tilde{z} which is larger than α (as j is huge, it cannot fully cover $\alpha\beta^{j/2}$). By extension (Lemma 4.1), we may write $r_i = (u_1, v_1; u_2, \tilde{v}\alpha\beta^\ell)$ for some $\ell \geq 1$. By Lemma 4.2 and as $\alpha \sim \alpha\beta$, we may replace this rule by $(u_1, v_1; u_2, \tilde{v}\alpha)$. Note that, as the forth component got shorter, now $r_i \in S$. Moreover, after we symmetrically treated rules of form (ii), these new rules r_1, \dots, r_k and the words w_1, \dots, w_k can be used in order to create $w = x_{k+1}zy_{k+1}$ from x_1zy_1 by splicing. In order to see this, you should first observe that even though the factors $\alpha\beta\gamma$ in z , which we pumped up before, may overlap with each other, the left-most (and right-most) position where we replaced β by β^j is preceded by the factor α (resp., succeeded by the factor γ) in \tilde{z} .

By contradiction, suppose $r_i \notin S$ for some i and, by symmetry, suppose this i -th splicing is of form (i). Thus, the forth component of r_i has to be $\mu = \delta_1\alpha\beta\gamma\delta_2$. As $|\delta_1| \geq 5m > |x_i|$, $\alpha\beta\gamma$ is a factor of \tilde{z} . The pumping algorithm ensured that (a) the prefix α is succeeded by $\beta^{j/2}$ or (b) the suffix γ is preceded by $\beta^{j/2}$. As $j/2$ is huge and the splicing position is too close to the left end of z_i , (b) is not possible. Thus, the forth component of r_i overlaps in more than $|\alpha|$ letters with the left-most factor $\alpha\beta^{j/2}$ in \tilde{z} and we used the replacement above, which ensured $r_i \in S$.

Therefore, if x_1zy_1 was in $L(I, S)$, then $w \in L(I, S)$ as well, which would contradict the choice of w . If $z_1 = x_1\tilde{z}y_1 \in I$, then x_1zy_1 , which is at most as long as z_1 , would belong to I and we are done.

Henceforth, we may assume $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ and the splicing position of this splicing lies within \tilde{z} . Let $s = (u, v_1; u_2, v)$, $\tilde{w}_1 = xuv_1$ and $\tilde{w}_2 = u_2vy$ where $|v_1|, |u_2| < m$, by Lemma 4.3 (here, x and y are newly chosen words). We have

$$z_1 = x_1\tilde{z}y_1 = xuvy$$

where xu is a proper prefix of $x_1\tilde{z}$ and vy is a proper suffix of $\tilde{z}y_1$. If $s \notin S$, then $u = \mu$, or $v = \mu$.

We will see next that if $u = \mu$ or $v = \mu$, then we can use a rule $\tilde{s} \in S$ and maybe slightly modified words in order to obtain z_1 by splicing. Suppose $u = \mu = \delta_1\alpha\beta\gamma\delta_2$. Thus, $\alpha\beta\gamma$ is a factor of \tilde{z} , as $|\delta_1| \geq 5m > |x_1|$, and, as such, either (a) α is succeeded by $\beta^{j/2}$ or (b) γ is preceded by $\beta^{j/2}$. If (b) holds, $\delta_1\alpha$ is a suffix of a word in β^+ . We may write $\delta_1\alpha = \beta_2\beta^\ell$ where $\ell \geq 0$ and β_2 is a suffix of β . Replace u by $\beta_2\gamma\delta_1$ and use this new rule \tilde{s} in order to splice $(\tilde{w}_1, \tilde{w}_2) \vdash_{\tilde{s}} z_1$. Note that the first component is now shorter than μ .

Otherwise, (a) holds and $\gamma\delta_2v$ is a prefix of a word in β^+ . As j is huge and γ is a prefix of a word in β^+ , we may extend v (Lemma 4.1) such that we can write $\gamma\delta_2 = \beta^{\ell_1}\beta_1$ and $v = \beta_2\beta^{\ell_2}\gamma$ where $\ell_1, \ell_2 \geq 0$ and $\beta_1\beta_2 = \beta$. Now, we pump down one of the β in the first component and β^{ℓ_2} in the forth component and we let $\tilde{s} = (\delta_1\alpha\beta^{\ell_1}\beta_1, v_1; u_2, \beta_2\gamma) \sim s$. As both components are shorter than μ , we see that $\tilde{s} \in S$ and

$$(x\delta_1\alpha\beta^{\ell_1}\beta_1v_1, u_2\beta_2\beta^{\ell_2+1}\gamma y) \vdash_{\tilde{s}} z_1,$$

i. e., we have shifted one β from \tilde{w}_1 to \tilde{w}_2 . Note that $\beta_2\gamma$ is a prefix of $\beta_2\beta^{\ell_2+1}\gamma$.

Treating the forth component analogously, justifies the assumption that $s \in S$.

Next, we will pump down the factors $\alpha\beta^j\gamma$ to $\alpha\beta\gamma$ in \tilde{z} again. At every position where we pumped up before, we are now pumping down (in reverse order) in order to obtain the words $\hat{x}, \hat{u}, \hat{v}, \hat{y}$ from the words x, u, v, u , respectively. For each pumping step:

If u is covered by the factor $\alpha\beta^j\gamma$ (which we pump down in this step), extend u to the left such that it becomes a prefix of $\alpha\beta^j\gamma$. Symmetrically, if v is covered by the factor $\alpha\beta^j\gamma$, extend v to the right such that it becomes a suffix of $\alpha\beta^j\gamma$ (Lemma 4.1). Observe that extension ensures, that the factor $\alpha\beta^j\gamma$ is covered by either xu , uv , or vy .

If $\alpha\beta^j$ or $\beta^j\gamma$ is fully covered by one of x, u, v , or y , then replace this factor by $\alpha\beta$ or $\beta\gamma$, respectively. Otherwise, we exemplarily show how to pump when $\alpha\beta^j\gamma$ is covered by xu . We can factorize

$$x = \tilde{x}\alpha\beta^{j_1}\beta_1 \qquad u = \beta_2\beta^{j_2}\gamma\tilde{u}$$

where $\beta_1\beta_2 = \beta$ and $j_1 + j_2 + 1 = j$. The pumping result are the words $\tilde{x}\alpha\beta_1$ and $\beta_2\gamma\tilde{u}$, respectively.

Observe that, after reversing all pumping steps, $\hat{x}\hat{u} \sim xu$, $\hat{v}\hat{y} \sim vy$, $\hat{x}\hat{u}\hat{v}\hat{y} = x_1zy_1$, and the rule $t = (\hat{u}, v_1; u_2, \hat{v})$ respects L . Furthermore, if we used extension for u (or v) in one of the steps, then $|\hat{u}| \leq m^2$ (resp., $|\hat{v}| \leq m^2$). No matter whether we used extension, $t \in S$. Recall that w was chosen as the shortest word from $L \setminus L(I, S)$. As $|\hat{x}\hat{u}v_1|, |u_2\hat{v}\hat{y}| < |z| + 6m = |w|$, the words $\hat{x}\hat{u}v_1$ and $u_2\hat{v}\hat{y}$ belong to $L(I, S)$, and as $(\hat{x}\hat{u}v_1, u_2\hat{v}\hat{y}) \vdash_t x_1zy_1$, we conclude that x_1zy_1 as well as w belong to $L(I, S)$ — the desired contradiction. \square

We can now prove our main result.

Proof of Theorem 4.7. Recall that for a splicing language L with $m = |M_L|$ we intend to prove that the splicing system (I, R) with $I = \Sigma^{<m^2+6m} \cap L$ and

$$R = \left\{ r \in \Sigma^{<m^2+10m} \times \Sigma^{<2m} \times \Sigma^{<2m} \times \Sigma^{<m^2+10m} \mid r \text{ respects } L \right\}$$

generates the language $L = L(I, R)$.

Obviously, $L(I, R) \subseteq L$. By Lemma 4.8, we may assume that L is generated by a splicing system (J, S) where

$$S = \left\{ r \in (\Sigma^{<m^2+10m})^4 \mid r \text{ respects } L \right\}.$$

In order to prove $L \subseteq L(I, R)$, we use induction on the length of words in L . For $w \in L$ with $|w| < m^2 + 6m$, by definition, $w \in I \subseteq L(I, R)$.

Now, consider $w \in L$ with $|w| \geq m^2 + 6m$. The induction hypothesis states that every word $\tilde{w} \in L$ with $|\tilde{w}| < |w|$ belongs to $L(I, R)$. Factorize $w = x\alpha\beta\gamma\delta y$ such that $|x|, |y| = 3m$, $|\alpha\beta\gamma| = m^2$, $\beta \neq \varepsilon$, $\alpha \sim \alpha\beta$, and $\gamma \sim \beta\gamma$.

Choose j huge ($j > |w| + m^2 + 10m$ and J does not contain words of length j or more). We let $z = \alpha\beta^j\gamma\delta$ and investigate the creation of $xzy \in L$ by splicing in (J, S) . As z is not a factor of a word in J , we can trace back the creation of xzy by splicing to the point where the factor z is affected for the last time. Let $z_{k+1} = x_{k+1}zy_{k+1}$, where $x_{k+1} = x$ and $y_{k+1} = y$, be created by k splittings from a word $z_1 = x_1zy_1$ which is created by a splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1$ with $\tilde{w}_1, \tilde{w}_2 \in L$, $s \in S$, and the splicing position lies in the factor z . Furthermore, for $i = 1, \dots, k$ the intermediate splittings are either

- (i) $(w_i, z_i) \vdash_{r_i} x_{i+1}zy_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in S$, $y_{i+1} = y_i$, and the splicing position lies to the left of the factor z or
- (ii) $(z_i, w_i) \vdash_{r_i} x_{i+1}zy_{i+1} = z_{i+1}$, $w_i \in L$, $r_i \in S$, $x_{i+1} = x_i$, and the splicing position lies to the right of the factor z .

As $|z| \geq m^2 + 10m$ we can apply Lemma 4.6. Thus, $w_1, \dots, w_k \in I$, $r_1, \dots, r_k \in R$, and $|x_1|, |y_1| < 5m$.

Consider a rule r_i in a splicing of form (i). Suppose the forth component of r_i covers a prefix of the factor $\alpha\beta^j$ in z which is larger than $\alpha\beta$ (as j is huge, it cannot fully cover $\alpha\beta^j$). We may write $r_i = (u_1, v_1; u_2, \tilde{v}\alpha\beta^\ell\beta_1)$ for some $\ell \geq 1$ and a prefix β_1 of β . By Lemma 4.2 and as $\alpha \sim \alpha\beta$, we may replace this rule by $(u_1, v_1; u_2, \tilde{v}\alpha\beta_1) \in R$. Moreover, after we symmetrically treated rules of form (ii), these new rules r_1, \dots, r_k and the words w_1, \dots, w_k can be used in order to create $w = x_{k+1}\alpha\beta\gamma\delta y_{k+1}$ from $x_1\alpha\beta\gamma\delta y_1$ by splicing. Thus, if $x_1\alpha\beta\gamma\delta y_1$ belongs to $L(I, R)$, so does w .

Now, consider the first splicing $(\tilde{w}_1, \tilde{w}_2) \vdash_s z_1 = x_1zy_1$. By Lemma 4.3, let $s = (u, v_1; u_2, v)$ such that $\tilde{w}_1 = xuv_1$, $\tilde{w}_2 = u_2vy$ and $|v_1|, |u_2| < m$ (here, x and y are newly chosen words). Hence,

$$z_1 = xuvy = x_1zy_1 = x_1\alpha\beta^j\gamma\delta y_1$$

where xu is a proper prefix of x_1z and vy is a proper suffix of zy_1 .

Next, we will pump down the factor $\alpha\beta^j\gamma$ to $\alpha\beta\gamma$ in z again in order to obtain the words $\hat{x}, \hat{u}, \hat{v}, \hat{y}$ from the word x, u, v, y , respectively. The pumping is done as in the proof of Lemma 4.8:

If u is covered by the factor $\alpha\beta^j\gamma$, extend u to the left such that it becomes a prefix of $\alpha\beta^j\gamma$. Symmetrically, if v is covered by the factor $\alpha\beta^j\gamma$, extend v to the right such that it becomes a suffix of $\alpha\beta^j\gamma$ (Lemma 4.1). The extension ensures that the factor $\alpha\beta^j\gamma$ is covered by either xu , uv , or vy .

If $\alpha\beta^j$ or $\beta^j\gamma$ is fully covered by one of x, u, v , or y , then replace this factor by $\alpha\beta$ or $\beta\gamma$, respectively. Otherwise, we exemplarily show how to pump when $\alpha\beta^j\gamma$ is covered by xu . We can factorize

$$x = \tilde{x}\alpha\beta^{j_1}\beta_1 \qquad u = \beta_2\beta^{j_2}\gamma\tilde{u}$$

where $\beta_1\beta_2 = \beta$ and $j_1 + j_2 + 1 = j$. The pumping result are the words $\hat{x} = \tilde{x}\alpha\beta_1$ and $\hat{u} = \beta_2\gamma\tilde{u}$, respectively.

Observe that, $\hat{x}\hat{u} \sim xu$, $\hat{v}\hat{y} \sim vy$, $\hat{x}\hat{u}\hat{v}\hat{y} = x_1\alpha\beta\gamma\delta y_1$, and the rule $t = (\hat{u}, v_1; u_2, \hat{v})$ respects L . Furthermore, if we used extension for u (or v), then $|\hat{u}| \leq m^2$ (resp., $|\hat{v}| \leq m^2$). No matter whether we used extension, $t \in R$. As $|\hat{x}\hat{u}v_1|, |u_2\hat{v}\hat{y}| < |z| + 6m = |w|$ and by induction hypothesis, the words $\hat{x}\hat{u}v_1$ and $u_2\hat{v}\hat{y}$ belong to $L(I, S)$. We conclude $x_1\alpha\beta\gamma\delta y_1 \in L(I, R)$ and, therefore, $w \in L(I, R)$ as well. \square

It is known how to decide whether a rule respects a regular language, e. g., see [4, 8]. Furthermore, there is an effective construction of a finite automaton which accepts the language generated by a splicing system [10]. Together with these two results, Theorem 4.7 answers the decidability question.

Corollary 4.9. *For a given regular language L , it is decidable whether L is a splicing language. Moreover, if L is a splicing language, a splicing system (I, R) generating L can be effectively constructed.*

A Pumping Algorithm

Before we prove Lemma A.1, let us recall a basic fact about primitive words. A word p is called *primitive* if there is no word x and $i \geq 2$ such that $p = x^i$. The *primitive root* of a word $w \neq \varepsilon$ is the unique primitive word p such that $w = p^i$ for some $i \geq 1$. For primitive p , it is well known that if $pp = xpy$, then either $x = p$ and $y = \varepsilon$ or $x = \varepsilon$ and $y = p$. Informally speaking, whenever p is a factor of a word in p^+ , then it has to match one of the ps .

Lemma A.1. *Let z, α, β, γ be words with $\beta \neq \varepsilon$ and $j > |z| + |\alpha\beta\gamma|$ be an even number. The following algorithm will terminate, eventually.*

1. let $\tilde{z} := z$;
2. if there is a factor $\alpha\beta\gamma$ of \tilde{z} such that neither
 - (a) its prefix α is succeeded by $\beta^{j/2}$ nor
 - (b) its suffix γ is preceded by $\beta^{j/2}$,
 then replace this factor by $\alpha\beta^j\gamma$;
3. repeat step 2 until there is no such factor $\alpha\beta\gamma$ left.

Proof. Let p be the primitive root of β and $\beta = p^k$.

First, observe that even though a factor $\alpha\beta\gamma$ for which neither (a) nor (b) holds may overlap with a factor $\alpha\beta^j\gamma$, it must not be fully covered by $\alpha\beta^j\gamma$ (or by a factor β^j). Indeed, if $\alpha\beta\gamma$ was fully covered by $\alpha\beta^j\gamma$, then, by symmetry, assume α is covered by $\alpha\beta^{j/2}$. Therefore, β is a prefix of $p^{k \cdot j/2} = \beta^{j/2}$ and (a) holds.

Let $z_0 = z$, let z_n be the word \tilde{z} after the n -th pumping step in the algorithm and let $y = p^{k \cdot j - 2}$ ($= \beta^j p^{-2}$). For each n , we will define a unique factorization

$$z_n = x_{n,0} y x_{n,1} \cdots y x_{n,n}$$

where p is a suffix of $x_{n,i}$ for $i = 0, \dots, n-1$ and p is a prefix of $x_{n,i}$ for $i = 1, \dots, n$. This factorization is defined inductively: Naturally, we start with $x_{0,0} = z_0 = z$. Now, let $\alpha\beta\gamma$ be the factor of z_n where neither (a) nor (b) holds that we replace in the $(n+1)$ -st step (if there is no such factor, the algorithm terminates). By contradiction, assume that its prefix α is covered by the i -th factor $y = p^{k \cdot j - 2}$ in the factorization of z_n for some $1 \leq i \leq n$. By the first observation we made, $\beta\gamma$ must overlap with x_i . However, as p is a prefix of x_i , the prefix p of β has to match with one of the p s, α is a suffix of a word in p^+ , and (b) holds — contradiction. Symmetrically, γ is not covered by one of the factors y either.

Thus, β is covered by some $x_{n,i}$, which we may factorize $u\beta v$ where $u \neq \varepsilon$ and $v \neq \varepsilon$. Note that the length of $x_{n,i}$ has to be at least $|\beta| + 2$. Now, let $x_{n+1,\ell} = x_{n,\ell}$ for $\ell = 0, \dots, i-1$, let $x_{n+1,\ell+1} = x_{n,\ell}$ for $\ell = i+1, \dots, n$, let $x_{n+1,i} = up$ and $x_{n+1,i+1} = pv$. Observe that this defines the desired factorization. Also note that

$$|x_{n+1,i}| = |u| + |p| = |x_{n,i}| - |\beta| - |v| + |p| \leq |x_{n,i}| - |v| < |x_{n,i}|$$

and, symmetrically, $|x_{n+1,i+1}| < |x_{n,i}|$. Thus, in each pumping step, we replace one of the factors $x_{n,i}$ by two strictly shorter factors $x_{n+1,i}$ and $x_{n+1,i+1}$. As we have noted above, in a factor $x_{n,i}$ cannot be pumped anymore, if it is shorter than $|\beta| + 2$. Eventually, all the factors will be too short and the pumping algorithm will stop. \square

References

- [1] K. Culik II and T. Harju. Splicing semigroups of dominoes and DNA. *Discrete Applied Mathematics*, 31(3):261–277, 1991.
- [2] K. L. Denninghoff and R. W. Gatterdam. On the undecidability of splicing systems. *International Journal of Computer Mathematics*, 27(3-4):133–145, 1989.
- [3] R. W. Gatterdam. Splicing systems and regularity. *International Journal of Computer Mathematics*, 31(1-2):63–67, 1989.
- [4] E. Goode. *Constants and Splicing Systems*. PhD thesis, Binghamton University, 1999.

- [5] E. Goode and D. Pixton. Recognizing splicing languages: Syntactic monoids and simultaneous pumping. *Discrete Applied Mathematics*, 155(8):989–1006, 2007.
- [6] T. Head. Formal language theory and dna: an analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, 49(6):737–759, 1987.
- [7] T. Head. Splicing schemes and DNA. *Nanobiology.*, 1:335–342, 1992.
- [8] T. Head, D. Pixton, and E. Goode. Splicing systems: Regularity and below. In M. Hagiya and A. Ohuchi, editors, *DNA*, volume 2568 of *Lecture Notes in Computer Science*, pages 262–268. Springer, 2002.
- [9] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
- [10] D. Pixton. Regularity of splicing languages. *Discrete Applied Mathematics*, 69(1-2):101–124, 1996.
- [11] G. Păun. On the splicing operation. *Discrete Applied Mathematics*, 70(1):57 – 79, 1996.