# Developments on circular splicing *

Paola Bonizzoni †, Clelia De Felice**, Giancarlo Mauri †, Rosalba Zizza †

## 1 Introduction

In 1987 Tom Head introduced a new operation on strings, called *splicing*, as a formal model of certain cut and paste biochemical transformation's processes of an initial collection of DNA strands under the simultaneous influence of enzymes [9]. Two strands of DNA are cut at a specified substrings (sites) by restriction enzymes that recognize a pattern inside the molecule and then the fragments are pasted by ligase enzymes. Since then, his basic idea has been formalized in terms of generative mechanisms for formal languages, called *splicing systems* or, more recently, $H$ systems. An $H$ system is defined by giving an initial language $I$ (initial set of DNA molecules) and a set of special words or rules $R$ (enzymes). The set $I$ is then transformed by repeated applications of splicing operation.

Intensive studies on splicing systems and variants have been done, oriented to prove closure properties or universality of splicing languages, generated under different restrictions on $I$ and/or $R$ [16, 21]. Regularity of splicing languages generated by regular initial languages and finite set of rules, characterizations of various classes of splicing languages and results on the recursive enumerability of languages are given and collected in [11, 17]. A complete set of references with respect to the relation between regularity and splicing may be found in [4].

On the contrary, a relatively few works on circular splicing systems and their languages have been published. Indeed, Tom Head reviewed a previous work on splicing systems and strengthened the significance of studying circular splicing and mixed splicing (both linear and circular words are allowed). Thus, in [23], the authors proposed several splicing operations involving circular strings and showed some examples of not circular regular languages generated by finite circular splicing (i.e., with $I, R$ both finite sets). Later, Pixton proved closure properties of circular splicing by using two different definitions of splicing [11, 19, 20]. In particular, Pixton proved that a circular regular language and a finite set of rules generate a regular language if the set of rules satisfies some additional hypothesis [21]. A similar closure property of splicing has been proved under a more general hypothesis on $I$ (full AFL) and for Paun systems [19, 20].

Even if pratical experiments on circular DNA have been very recently and successfully done [10, 12], we want to emphasize that our approach is purely language-theoretic oriented. In fact, in this paper we carry on the investigation initiated in [6, 3] and [4], where some results are given about the computational power of finite (Paun) circular splicing systems, with no additional hypothesis. More precisely, since it is well-known that finite circular splicing does not generate only circular regular languages, we focus our interest on which kind of regular circular

*Partially supported by 60% Project *"Linguaggi Formali e Modelli di Calcolo"* (University of Salerno)

†Dipartimento di Informatica Sistemistica e Comunicazione, Università degli Studi di Milano – Bicocca, Via Bicocca degli Arcimboldi 8, 20126 Milano – ITALY. E-mail:{bonizzoni,mauri,zizza}@disco.unimib.it

** Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi (SA) – ITALY. E-mail: defelice@unisa.it.

languages are generated by circular splicing with both finite $I$ and $R$, a problem which was directly approached in [6, 3] and [4] for the first time.

All the necessary preliminary definitions on circular languages and circular splicing operation are collected in Section 2. Sections 3 - 6 present developments on splicing, answering some questions posed in [4], recalled here. Precisely, in [4] the authors gave a characterization of (circular) regular languages on one-letter alphabet generated by a finite Paun circular splicing system. In Section 3 we prove that we can decide whether a regular language on one-letter alphabet is generated by a finite Paun circular splicing system. Furthermore, we generalize this result, characterizing circular regular languages $L = A^J = \bigcup_{j \in J} A^j$, $J$ a subset of $\boldsymbol{N}$. Section 5 presents a circular regular language which is neither Paun nor Pixton generated, proving that the class of circular regular splicing languages is a proper subset of the class of all circular regular languages. Finally, we introduce a class of circular regular languages which are generated by circular Pixton splicing.

All the proofs which are not contained in this paper can be found in a manuscript which is available upon request to the authors [5].

## 2 Basics on circular (splicing) languages

Let us fix some notations. We denote by $A^*$ the free monoid over a finite alphabet $A$, by $|w|$ the length of $w \in A^*$ and we set $A^+ = A^* \setminus 1$, where 1 is the empty word. For every $a \in A$, $w \in A^*$, we denote $|w|_a$ the number of occurrences of $a$ in $w$, whereas for a subset $X$ of $A^*$, we set $|w|_X = \sum_{x \in X} |w|_x$ and $|X|$ the cardinality of $X$. In the following we denote by $\mathcal{A} = (Q, A, \delta, q_0, F)$ a trim deterministic automaton recognizing a regular language $L = L(\mathcal{A})$. As usual, in the transition diagram of a trim deterministic automaton $\mathcal{A}$, each finite state will be indicated by a double circle and the initial state will be indicated by an arrow without label going in it.

In order to present circular splicing, we need some definitions. Circular words have already been intensively examined in formal language theory (see [1, 2, 22]). For a given word $w \in A^*$, a circular word $\sim w$ is the equivalence class of $w$ with respect to the *conjugacy* relation $\sim$ defined by $xy \sim yx$, for $x, y \in A^*$ (see [14]). For every $a \in A$, the notations $|\sim w|$ and $|\sim w|_a$ will be defined respectively as $|w|$ and $|w|_a$, for any representative $w$ of $\sim w$. Furthermore, when the context does not make it ambiguous, we will use the notation $w$ for a circular word $\sim w$. $\sim A^*$ is the set of all circular words over $A$, i.e. the quotient of $A^*$ with respect to $\sim$. Given $L \subseteq A^*$, $\sim L = \{\sim w \mid w \in L\}$ is the *circularization* of $L$, i.e. the set of all circular words corresponding to elements of $L$, while every language $L$ such that $\sim L = C$, for a given circular language $C \subseteq \sim A^*$, is called a *linearization* of $C$. The set of all strings in $A^*$ corresponding to elements of $C$, is the *full linearization* of $C$, written $Lin(C)$.

Observe that circular splicing deals with circular strings and circular languages and as a result, with formal languages which are full linearizations of circular languages. Thus, in the next part of this paper, we will consider languages closed under the conjugacy relation and the action of circular splicing over their circularization.

We denote $Fin$ (resp. $Reg$) the class of finite (resp. regular) languages. Given a family of languages $FA$ in the Chomsky hierarchy, $FA^\sim$ consists of all those circular languages $C$ which have some linearization in $FA$. In this paper we mainly deal with $Reg^\sim$. It is classically known that given a regular language $L \subseteq A^*$, its circularization $\sim L$ is regular if and only if its full linearization $Lin(\sim L) = \{w' \in A^* \mid \exists w \in L.\ w' \sim w\}$ is regular [13].

The circular splicing operation has been modelled by three definitions, given by Head, Paun and Pixton. In this paper, we restrict ourselves to Paun and Pixton definitions since we are mainly concerned with them.

`Paun's definition [19, 11].` A *Paun circular splicing system* is a 3-tuple $SC_{PA} = (A, I, R)$, where $I \subseteq^{\sim} A^*$ is the initial circular language and $R \subseteq A^*|A^*\ \$A^*|A^*$, with $|, \$ \notin A$, is the set of rules. Then, given a rule $r = u_1|u_2\$u_3|u_4$ and two circular words $^{\sim}u_2hu_1, ^{\sim}u_4ku_3$, the rule cuts and linearizes the two strings obtaining $u_2hu_1$ and $u_4ku_3$, and pastes and circularizes them obtaining $^{\sim}u_2hu_1u_4ku_3$.

`Pixton's definition [21].` A *Pixton circular splicing system* is a 3-tuple $SC_{PI} = (A, I, R)$ where $A$ is a finite alphabet, $I \subseteq^{\sim} A^*$ is the initial circular language, $R \subseteq A^* \times A^* \times A^*$ is the set of rules. $R$ is such that if $r = (\alpha, \alpha'; \beta) \in R$ then there exists $\beta'$ such that $\overline{r} = (\alpha', \alpha; \beta') \in R$. Thus, given two circular words $^{\sim}\alpha\epsilon, ^{\sim}\alpha'\epsilon'$, the two rules $r, \overline{r}$ cut and linearize the two strings, obtaining $\epsilon\alpha, \epsilon'\alpha'$ and then paste, substitute and circularize them, producing $^{\sim}\epsilon\beta\epsilon'\beta'$.

We now give the definition of circular splicing languages. For a given splicing system $SC_X$, with $X \in \{PA, PI\}$, we denote $(w', w'')\vdash_r w$ the fact that $w$ is produced from (or spliced by) $w', w''$ by using a rule $r$. Furthermore, given a language $C \subseteq^{\sim} A^*$, we denote $\sigma'(C) = \{z \in^{\sim} A^* \mid \exists w', w'' \in C, \exists r \in R.\ (w', w'')\vdash_r z\}$. Thus, we define $\sigma^0(C) = C$, $\sigma^{i+1}(C) = \sigma^i(C) \cup \sigma'(\sigma^i(C))$, $i \geq 0$, and then $\sigma^*(C) = \bigcup_{i\geq 0} \sigma^i(C)$. We explicitly note that for Pixton's systems, the splicing operation is the combined action of a pair of rules, $r$ and $\bar{r}$ and we can use the notation $\vdash_{r,\bar{r}}$ to make this behaviour evident.

**Definition 2.1** *Given a splicing system $SC_X$, with $X \in \{PA, PI\}$, the circular language $C(SC_X) = \sigma^*(I)$ is the language generated by the system $SC_X$, $I$ being the initial language in $SC_X$.*
*A circular language $C$ is $C_X$ generated (or $C$ is a circular splicing language) if a splicing system $SC_X$ exists such that $C = C(SC_X)$.*

In this paper we only deal with finite circular splicing systems, i.e., with both $I, R$ finite sets. The corresponding class of generated languages will be denoted $C_X(Fin, Fin)$, where $X \in \{PA, PI\}$. We must note that in the original definition of circular splicing language given by Paun in [11], rules in $R$ can be used in two different ways: one way has been described above, the other, called *self-splicing* will not be considered here. Furthermore, as we have already said, additional hypotheses can be added to the definitions of circular splicing, but we will not take them into account.

We conclude this section with Proposition 2.1. This proposition shows that the computational power of circular splicing systems increases when we substitute Head's systems with Paun's systems and Paun's systems with Pixton's systems.

**Proposition 2.1** *[3] If $C \subseteq^{\sim} A^*$ is generated by a finite Head circular splicing system then $C$ is $C_{PA}$ generated. Moreover, if $C \subseteq^{\sim} A^*$ is $C_{PA}$ generated, then $C$ is $C_{PI}$ generated.*

## 3 A decision problem

We have reported in Proposition 3.1 a characterization of $Reg^{\sim} \cap C_{PA}(Fin, Fin)$ for a one-letter alphabet, proved in [4]. As already observed in the same paper, each language $L \subseteq a^*$ is closed under the conjugacy relation since we have $^{\sim}w = \{w\}$, so $L =^{\sim} L$, and we can identify each word (resp. language) with its circularization. We will also use some notations and elementary notions from group theory recalled below (see for instance [8]).

Consider the *cyclic group* of order $n$ which we realize, as usual, as the quotient group $\boldsymbol{Z}_n$ of the integers modulo $n$ (i.e., the group of the congruence classes modulo $n$). For subsets $T_1, T_2, T$

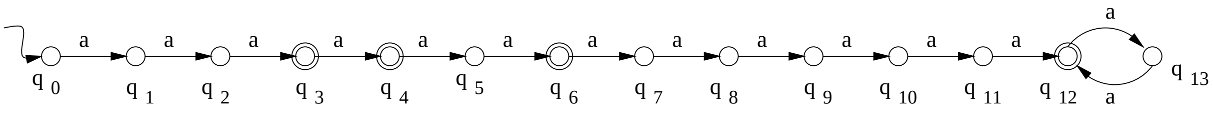


Figure 1: Automaton for $L = \{a^3, a^4\} \cup \{a^6, a^{14}, a^{16}\}^+$.

of $\boldsymbol{N}$ and $n \in \boldsymbol{N}$, $T_1 + T_2 = T \pmod n$ means that for any $t \in T$ a unique pair $(t_1, t_2)$ exists with $t_1 \in T_1, t_2 \in T_2$ and $t_1 + t_2 = t \pmod n$. When $T = \{0, \ldots, n-1\} = \boldsymbol{Z}_n$, the pair $(T_1, T_2)$ is called a *factorization* of $\boldsymbol{Z}_n$ (see [8] for definitions and main results and [7] for a survey on this argument). $T$ is a *periodic* subset of $\boldsymbol{Z}_n$ if $t \in \boldsymbol{Z}_n$ exists with $t + T = T \pmod n$. Each subgroup $G$ of $\boldsymbol{Z}_n$ is periodic (since $0 \in G$) but periodic subsets exist which are not subgroups. For instance, $\{0, 2, 4\}$ and $\{1, 3, 5\}$ are both periodic subsets of $\boldsymbol{Z}_6$ ($\{0, 2, 4\} + 2 = \{0, 2, 4\} \pmod 6$ and $\{1, 3, 5\} + 2 = \{1, 3, 5\} \pmod 6$), but the former is a subgroup of $\boldsymbol{Z}_6$ whereas the latter is not a subgroup ($5 + 1 \not\in \{1, 3, 5\}$).

For a subset $L$ of $A^*$ we denote $Fact(L)$ the set of all the factors of the elements of $L$.

**Proposition 3.1** *[4] A subset $L = {}^{\sim}L$ of $a^*$ is $C_{PA}$ generated if and only if there exist a finite subset $L_1$ of $a^*$, a positive integer $n$ and a (periodic) subgroup $G'$ of $\boldsymbol{Z}_n$ such that $L = L_1 \cup (a^G)^+$, where $G$ is a set of representatives of the elements in $G'$ and $a^G \cap Fact(L_1) = \emptyset$. Furthermore, $G = \{m + \lambda_m n \mid m \in G'\}$ with $\lambda_m = min\{\mu \mid a^{m+\mu n} \in (a^G)^+\}$.*

A decidability question naturally follows on from this result: given a language $L$, with $L \subseteq a^*$, can we decide whether $L$ is a $C_{PA}$ generated language? If no hypothesis is made over $L$, the answer to this question is no, thanks to the Rice theorem [13]. On the contrary, in this section, we prove that the answer to this question is yes when we restrict ourselves to regular languages. In other words, we prove that the following question is decidable.

**Problem 3.1** *Given a regular language $L$, with $L \subseteq a^*$, can we decide whether $L$ is a $C_{PA}$ generated language?*

As a matter of fact, we will suppose that $L \subseteq a^*$ is given by means of a determinimistic finite state automaton $\mathcal{A}$, which will be the instance of our Problem 3.1. Thus, in Proposition 3.2 we give four conditions: we must check that $\mathcal{A}$ satisfies these conditions in order to state that the language $L(\mathcal{A})$ recognized by the automaton $\mathcal{A}$ is a $C_{PA}$ generated language.

Given a regular language $L \subseteq a^*$, well-known results describe the structure of every deterministic finite state automaton $\mathcal{A}= (\{a\}, \{q_0, q_1, \ldots, q_v\}, \delta, q_0, F)$ recognizing $L$. The transition function $\delta$ is defined as follows: for each $i \in \{0, \ldots v-1\}$ we have $\delta(q_i, a) = q_{i+1}$ and there exists $l \in \{0, \ldots, v-1\}$ such that $\delta(q_v, a) = q_l$. In other words, the transition graph of $\mathcal{A}$ has a (frying-)pan shape: the path crossing the states $q_0, \ldots, q_l$ is the handle of the pan, the closed path crossing the states $q_l, \ldots, q_v$ is the pan without the handle. As an example consider the regular $C_{PA}$ generated language $L = \{a^3, a^4\} \cup \{a^6, a^{14}, a^{16}\}^+$. Two deterministic finite state automata recognizing $L$ are depicted in Figure 1 and Figure 2.

We only glance at the conditions reported in Proposition 3.2. Intuitively, in any automaton $\mathcal{A}$ recognizing a $C_{PA}$ generated language $L = L(\mathcal{A})$ the distance between any pair of consecutive final states on the pan without the handle does not depend on the chosen pair. Furthermore, if at least one of these pairs exists, say $p$ their distance. Otherwise say $p$ the length of the closed path in $\mathcal{A}$. Then, there exists a final state $q_n$ on the handle such that, when we consider the set of the indices (modulo $n$) of the final states from $q_n$ to $q_v$, this set is a subgroup of $\boldsymbol{Z}_n$ of period $p$. This set must also satisfy two supplementary conditions.

In the following result we have reported the well known structure of the subgroups of $\boldsymbol{Z}_n$, used in the proof of Proposition 3.2.

**Lemma 3.1** *Let $G'$ be a subgroup of $\boldsymbol{Z}_n$ of period $p$. Then, set $n = pr$, we have $G' = \{0 + pk \mid 0 \leq k \leq r-1\}$. Furthermore, for each $t \in G'$, $G' = \{t + pk \mid 0 \leq k \leq r-1\} \pmod{n}$.*

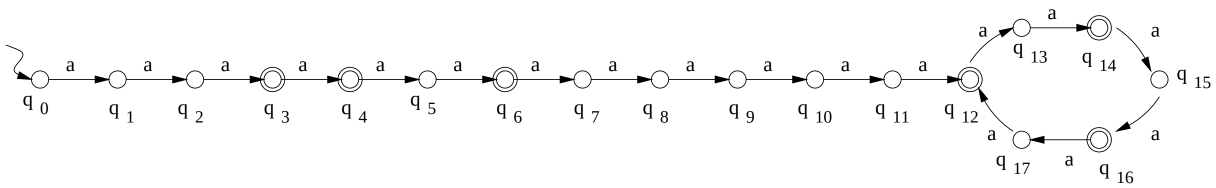


Figure 2: Another automaton for $L = \{a^3, a^4\} \cup \{a^6, a^{14}, a^{16}\}^+$

**Proposition 3.2** *Let $\mathcal{A}= (\{a\}, \{q_0, q_1, \ldots, q_v\}, \delta, q_0, F)$ be a deterministic finite state automaton where for each $i \in \{0, \ldots v-1\}$ we have $\delta(q_i, a) = q_{i+1}$ and $\delta(q_v, a) = q_l$ for $l \in \{0, \ldots, v-1\}$. Set $F = \{q_{i_1}, \ldots, q_{i_g}\} \cup \{q_{i_{g+1}}, \ldots, q_{i_h}\}$ with $i_1 < i_2 < \ldots < i_g < l \leq i_{g+1} < \ldots < i_h$. We have that $L = L(\mathcal{A})$ is $C_{PA}$-generated if and only if $\mathcal{A}$ satisfies the following four conditions.*

1. *There exists $p$, $p \in \boldsymbol{N}$, such that for all $q_{i_j} \in F$, $g+1 \leq j \leq h-1$, we have $i_{j+1} - i_j = p$ and $i_{g+1} + (v-l+1) - i_h = p$. (If $g+1 = h$, this condition vanishes and we simply set $v - l + 1 = p$).*
2. *There exists $n \in \{0, \ldots l\}$ such that $n = pr$, with $r \in \boldsymbol{N}$, $q_n \in F$ and the set $G' = (\{i_j \mid n \leq i_j < l\} \cup \{i_{g+1} + pk \mid 0 \leq k \leq r-1\}) \pmod{n}$ is a subgroup of $\boldsymbol{Z}_n$ of period $p$.*
3. *For each $m \in G'$, let $\lambda_m = min\{\mu \mid m+\mu n \in \{i_j \mid n \leq i_j < l\} \cup \{i_{g+1}+pk \mid 0 \leq k \leq r-1\}\}$ and set $G = \{m + \lambda_m n \mid m \in G'\}$. Then, we have $\lambda_m = min\{\mu \mid a^{m+\mu n} \in (a^G)^+\}$.*
4. *For each $m \in G'$ such that $m + \lambda_m n \in \{i_j \mid n \leq i_j < l\}$, for each $\mu \geq \lambda_m$ such that $m + \mu n < l$, we must have $m + \mu n \in \{i_j \mid n \leq i_j < l\}$.*

By using Proposition 3.2, we prove that we can decide whether a deterministic finite state automaton $\mathcal{A}$ satisfies the four conditions reported in Proposition 3.2 and so we have the following result.

**Proposition 3.3** *Given a regular language $L$, with $L \subseteq a^*$, we can decide whether $L$ is a $C_{PA}$ generated language.*

## 4 The case of uniform languages

In the previous section, we have seen that we can decide whether a regular language $L \subseteq a^*$ is $C_{PA}$-generated or not by means of properties of the corresponding automaton recognizing $L$. In this section, we will see how the characterization given in Proposition 3.1 can be easily generalized to languages $L$ with the form $L = \bigcup_{j \in J} A^j$, $A$ being an alphabet with $|A| \geq 1$ (Definition 4.1).

**Definition 4.1** *A language $L$ is an* uniform language *if there exists a subset $J$ of $\boldsymbol{N}$ such that $L = A^J = \bigcup_{j \in J} A^j = \bigcup_{j \in J} \{w \in A^* \mid |w| = j\}$.*

We recall that, given $j \in \boldsymbol{N}$, the set $A^j = \{w \in A^* | |w| = j\}$ is a code, called *uniform code*. Observe that $Lin(\sim(A^j)) = A^j$, i.e. we can identify $A^j$ with its circularization. Moreover, given $L, L' \in A^*$, $\sim(L \cup L') = \sim L \cup \sim L'$ (resp. $\sim(L \cap L') = \sim L \cap \sim L'$). This means that the application associating to each language its circularization is a morphism with respect to union and intersection. Thus, we have that $Lin(\sim A^J) = A^J$ and we can identify each language $A^J$ with its circularization.

Using Lemmata 4.1 and 4.2 as preliminary results, we give a characterization of all subsets $J$ of $\boldsymbol{N}$ such that $A^J \in Reg^\sim \cap C_{PA}(Fin, Fin)$. Observe that circular splicing systems with an initial finite set and an empty set $R$ of rules generates only finite sets. This particular situation is not significant for a characterization of $Reg^\sim \cap C_{PA}(Fin, Fin)$, so in the next part of this section we will often suppose $R \neq \emptyset$.

**Lemma 4.1** *Let $SC_{PA} = (A, I, R)$ be a circular splicing system such that there exists $J \subseteq \boldsymbol{N}$ with $C(SC_{PA}) = A^J$. Denote $R = \{r_1, \ldots, r_t\}$ and $r_i = w_{1,i}|w_{2,i}\$w_{3,i}|w_{4,i}$, where $w_{j,i} \in A^*$, for $j \in \{1,2,3,4\}$, $i \in \{1, \ldots, t\}$. For all $i \in \{1, \ldots, t\}$, for all $x, y \in A^J$ with $|x| \geq |w_{1,i}| + |w_{2,i}|$, $|y| \geq |w_{3,i}| + |w_{4,i}|$, there exist $x', y'$, $|x'| = |x|$ and $|y'| = |y|$ such that $\sim x'y'$ is generated by means of the rule $r_i$. Consequently, $\sim xy \in A^J = C(SC_{PA})$.*

**Lemma 4.2** *Let $SC_{PA} = (A, I, R)$ be a circular splicing system such that there exists $J \subseteq \boldsymbol{N}$ with $C(SC_{PA}) = A^J$. Denote $R = \{r_1, \ldots, r_t\}$ and $r_i = w_{1,i}|w_{2,i}\$w_{3,i}|w_{4,i}$, where $w_{j,i} \in A^*$, for $j \in \{1,2,3,4\}$, $i \in \{1, \ldots, t\}$. Let $m = max$ $\{|w_{1,i}| + |w_{2,i}|, |w_{3,i}| + |w_{4,i}| \mid i \in \{1, \ldots, t\}\}$ and $L_2 = \{w \in A^J \mid |w| \geq m\}$. Then, $A^J = L_1 \cup L_2$ where $L_1 = A^J \setminus L_2$ is a finite subset of $A^J$. Furthermore, either $L_2 = \emptyset$ or there exists $n \in \boldsymbol{N}$ such that $L_2 = (A^G)^+$, where $G$ is a set of representatives of the elements in a (periodic) subgroup of $\boldsymbol{Z}_n$ and $A^G \cap Fact(L_1) = \emptyset$.*

A characterization of the uniform languages with a $C_{PA}$ generated circularization is given in Proposition 4.2. This characterization is obtained also by using the result recalled below.

**Proposition 4.1** *[4] For all finite sets $X, Y$ such that $X^* \cup Y$ is closed under the conjugacy relation and with $X \cap Fact(Y) = \emptyset$, we have that $\sim(X^* \cup Y)$ (resp. $\sim(X^+ \cup Y)$) is a circular splicing language.*

**Proposition 4.2** *Given $J \subseteq \boldsymbol{N}$, a language $L = A^J$ is $C_{PA}$ generated if and only if there exist a finite subset $L_1$ of $A^*$, a positive integer $n$ and a (periodic) subgroup $G'$ of $\boldsymbol{Z}_n$ such that $L = L_1 \cup (A^G)^+$, where $G$ is a set of representatives of the elements in $G'$ and $A^G \cap Fact(L_1) = \emptyset$.*

Corollary 4.1 provides a simpler characterization of the uniform languages with a $C_{PA}$ generated circularization.

**Corollary 4.1** *Given $J \subseteq \boldsymbol{N}$, a language $L = A^J$ is $C_{PA}$ generated if and only if there exist finite subsets $L_1$, $A^G$ of $A^J$ and a positive integer $n$ such that $L = L_1 \cup (A^G)^+$ and, when $A^G \neq \emptyset$, $max\{|l| \mid l \in L_1\} < n = min\{|l| \mid l \in A^G\}$.*

We end this section with some observations concerning the descriptional complexity of a circular splicing system which generates a circular language $L = A^J$ (Proposition 4.3). Observe that we are not able to exhibit a minimal circular splicing system, as we have done for one-letter alphabets in [4].

**Proposition 4.3** *Let $L = A^J$ be a $C_{PA}$ generated language, $J \subseteq \boldsymbol{N}$. Then, there exists $n \in \boldsymbol{N}$ and a splicing system $SC_{PA} = (A, I, R)$ generating $L$ with either $R = \emptyset$ or $R$ containing at most $\frac{|A^n| \cdot (|A^n|+1)}{2}$ rules.*

**Example 4.1** Let $L = \{A^3, A^4\} \cup \{A^6, A^{14}, A^{16}\}^+$. Then $L$ is an uniform language which is $C_{PA}$ generated since $L$ satisfies the hypothesis contained in Proposition 4.2 with $n = 6$. Denote $A^6 = \{\alpha_i \mid i \in \{1, \ldots, |A^6|\}\}$. According to Proposition 4.3, $L = C(SC_{PA})$, where $SC_{PA} = (A, I, R)$, $I = \{A^3, A^4, A^6, A^{14}, A^{16}\}$ and $R = \{\alpha_i \mid 1 \$ 1 \mid \alpha_j; j \geq i, i \in \{1, \ldots, |A^6|\}\}$.

## 5 A circular regular language not Pixton generated

In this section we will consider the circular regular language $\sim((A^2)^* \cup (A^3)^*)$ and we are able to prove that this language cannot be generated by finite splicing, even in the case of the more powerful definition given by Pixton. The same result for Paun systems can be found in [4].

We preliminary observe that $Lin(\sim(A^n)^*) = (A^n)^*$ and $\sim(A^n)^*$ is a regular circular language, since its full linearization is regular (as a matter of fact this full linearization belongs to the class of star languages which are defined in the next section). Furthermore, if we take into account the observation made in the previous section, we have that $Lin(\sim((A^2)^* \cup (A^3)^*)) = (A^2)^* \cup (A^3)^*$, and $\sim((A^2)^* \cup (A^3)^*)$ is also a circular regular language. Let us sketch the proof of this result. By contradiction, suppose that there exists $SC_{PI} = (A, I, R)$ such that $L = \sim((A^2)^* \cup (A^3)^*) = C(SC_{PI})$ and consider $(\alpha, \alpha'; \beta)$, $(\alpha', \alpha; \beta') \in R$. We will distinguish the following four different cases with respect to $|\beta\beta'|$.

1. $|\beta\beta'| \equiv 0 \pmod 2$, $|\beta\beta'| \equiv 0 \pmod 3$;
2. $|\beta\beta'| \equiv 0 \pmod 2$, $|\beta\beta'| \not\equiv 0 \pmod 3$,
3. $|\beta\beta'| \not\equiv 0 \pmod 2$, $|\beta\beta'| \equiv 0 \pmod 3$;
4. $|\beta\beta'| \not\equiv 0 \pmod 2$, $|\beta\beta'| \not\equiv 0 \pmod 3$

By using combinatorial arguments, we show that in each of the above-mentioned four cases, we have $\sim((A^2)^* \cup (A^3)^*) \neq C(SC_{PI})$ and so we can state the following proposition.

**Proposition 5.1** *For every circular Pixton splicing system $SC_{PI}$, the circular regular language $\sim((A^2)^* \cup (A^3)^*) \neq C(SC_{PI})$.*

As we have already said, Proposition 5.1 shows that we cannot generate all circular regular languages by using finite circular splicing.

# 6 Regular $C_{PI}$ generated languages

In this section we define a class of regular languages whose circularization is $C_{PI}$ generated. Since we will often refer to a class of languages introduced in [4], we report here some definitions. In [4] the authors consider a class of languages $X^* \subseteq A^*$, called *star languages*, which are closed under conjugacy relation and with $X$ being a regular language. By using automata theory and combinatorial techniques on words, the authors show that for two subclasses of star languages the corresponding circular languages are (Paun) circular splicing languages. The first class is defined as the family of star languages $X^*$ with $X$ being a finite set. The second class, whose languages are called *fingerprint closed* star languages in [4], contains all star languages $X^*$ with $X^*$ being a free monoid (as a matter of fact we do not yet know of any examples of fingerprint closed star languages which do not satisfy this hypotheses).

Let us recall that an algebraic description of some subclasses of the class of the regular languages $L \subseteq A^*$ has been given by means of the *syntactic monoid* $\mathcal{M}(L)$ of $L$. This is the quotient of $A^*$ with respect to the syntactic congruence $\equiv_L$, defined as follows: $w \equiv_L w'$, with $w, w' \in A^*$, if and only if $xwy \in L \Leftrightarrow xw'y \in L$, for all $x, y \in A^*$. If $L$ is regular, a well known result states that $\mathcal{M}(L)$ is a finite monoid. Moreover $X^*$ is a free monoid if and only if $X$ is a *code*, i.e., for all $x_1, \ldots, x_n$, $x'_1, \ldots, x'_m \in X$, we have $x_1 \cdots x_n = x'_1 \cdots x'_m$ if and only if $n = m$ and $\forall i \in \{1, \ldots, n\}$ $x_i = x'_i$ [1]. In [2, 22] the authors prove that $X^* \subseteq A^*$ is a free monoid closed under conjugacy relation if and only if $\mathcal{M}(X^*)$ is a group. In [1, 2] it has been observed that, under the hypothesis that $X$ is a regular code, $\mathcal{M}(X^*)$ is a group if and only if $X$ is a *group code*, i.e., a group $G$ and a surjective morphism $\phi: A^* \to G$ exist such that $X^* = \phi^{-1}(H)$, where $H$ is a subgroup of $G$. For example, when $X$ is finite we have that $X \subseteq A^+$ is a group code if and only if there exists $d > 0$ such that $X$ is the uniform code $A^d$ [2, 22]. Thanks to Proposition 6.1, we have that star languages generated by group codes satisfy an additional hypothesis which allows us to state that they are $C_{PA}$ generated languages.

**Proposition 6.1** *[4] Let $X$ be a regular group code and let $\mathcal{A}$ be a trim deterministic finite state automaton recognizing $X^*$. For any cycle $c$ in $\mathcal{A}$, $c \in X^*$.*

In [4], a collection of languages are proved to be star languages generated by group codes (and then such that their circularization is $C_{PA}$ generated). For instance, $(aa)^*$ is a star language generated by the group code $\{aa\} = \{a\}^2$ and so $\sim(aa)^*$ is $C_{PA}$ generated. By using combinatorial arguments, in the same paper, the authors show that $(aa)^*b$, $\{w \in A^* \mid \exists h, k \in \boldsymbol{N}\ |w|_a = 2k+1, |w|_b = 2h+1\}$ and $A^* \setminus a^* = (a^*ba^*)^*$ are regular languages which are closed under conjugacy and whose circularizations cannot be $C_H$ or $C_{PA}$ generated but which are $C_{PI}$ generated. Analogously, one can prove that $cyclic(aa) = {}^\sim(a^2)^*a$, $cyclic(bb) = {}^\sim(b^2)^*b$, $cyclic(ab) = {}^\sim(ab)^*a \cup {}^\sim(ab)^*b = cyclic(ba)$ are regular circular languages not $C_H$ or $C_{PA}$ generated but $C_{PI}$ generated [4].

In this section we define a subclass of regular languages closed under the conjugacy relation whose circularization is Pixton generated. We will see that $A^* \setminus a^*$ belongs to this subclass (Proposition 6.7).

Given a trim deterministic automaton $\mathcal{A}$ recognizing $L$, in Definition 6.1, we recall a formalism for referring to some special labels $c$ (cycles) of the closed paths in $\mathcal{A}$ given in [4]. We then recall the definition of a (finite) subset of these special labels (Definition 6.2). Indeed, intuitively, if a rational not finite language $L$ is generated by splicing, we must exibit rules which are able to produce words with a non bounded number of occurrences of such labels as factors.

**Definition 6.1 (Cycle)** *A word $c \in A^+$ is a* cycle *(resp.* simple cycle*) in $\mathcal{A}$ if $q \in Q$ exists such that $\delta(q, c) = q$ and the internal states crossed by the transition are different from $q$ (resp. each other and with respect to $q$).*

*Then, if $c$ is a cycle in $\mathcal{A}$ and $c$ is not simple, there exist $u_1, \ldots, u_{k+1} \in A^*$, cycles $c_1, \ldots, c_k \in A^+$ and positive integers $p_1, \ldots, p_k$ such that $c = u_1 c_1^{p_1} u_2 \cdots u_k c_k^{p_k} u_{k+1}$, where each $u_i$ is the label of transitions in which no state is crossed twice and $u_1, u_{k+1} \neq 1$. Furthermore, if $i, j \in \{1, \ldots, k\}$ exist such that $c_i = c_j$ then $u_{i+1} \cdots u_j \neq 1$.*

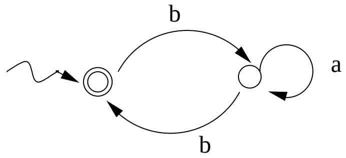


Figure 3: Automaton for $(ba^*b)^*$

**Example 6.1** Let us consider the regular language $L = (ba^*b)^*$ recognized by the automaton depicted in Figure 3. We have two cycles, $c_1 = a$ and $c_2 = ba^{p_1}b$, for $p_1 \geq 0$.

As already observed in [4], we are now faced with another problem. We explicitly notice that the property for $c$ of being a cycle of $\mathcal{A}$ is strictly depending on the automaton $\mathcal{A}$. Even if we restrict ourselves to a fixed automaton $\mathcal{A}$, we can have that $c$ is a cycle in $\mathcal{A}$ and $c$ is also the label of another (not necessary closed) path in $\mathcal{A}$. This is the reason for which below we will define a class of languages by means of an "aperiodicity condition" and a class $\mathcal{F}$ of languages by means of a "consistence condition" (see Definitions 6.7 and 6.6). Given a regular language $L$ which is closed under the conjugacy relation, we will see that we can define a Pixton splicing system $SC_{PI}$ such that $^\sim L \subseteq C(SC_{PI})$ even if we do not make any supplementary hypothesis on $L$ (Proposition 6.2). We will also see that the "consistence condition" is given to guarantee the consistence of the splicing system, i.e. $C(SC_{PI}) \subseteq {}^\sim L$ (Proposition 6.3). As a matter of fact, languages satisfying the "consistence condition" also satisfy the "aperiodicity condition". Nevertheless, the latter condition allows us to show that star languages generated by group codes belong to $\mathcal{F}$. However, $\mathcal{F}$ also contains other languages, as $A^* \setminus a^*$ shows (Proposition 6.7).

From now on $L$ will be a regular language $L$ which is closed under the conjugacy relation. We now recall the definition of the fingerprint of a cycle given in [4].

**Definition 6.2 (Fingerprint of a cycle)** *Let $\mathcal{A}$ be a finite state automaton recognizing a regular language $L$, i.e. $L = L(\mathcal{A})$. The set $\mathcal{C}(L) = \bigcup_{c \ cycle} F(c)$ of the fingerprints $F(c)$ for $L$ associated with $c$, for every cycle $c$ in $\mathcal{A}$, is inductively defined as follows:*

- *if $c$ is simple, then $F(c) = \{c\}$;*
- *if $c$ is not simple (i.e. $c = u_1 c_1^{p_1} u_2 c_2^{p_2} \cdots u_k c_k^{p_k} u_{k+1}$, where $u_i$ are labels of transitions in which no state is crossed twice, $u_1, u_{k+1} \neq 1$, $c_i$ are cycles, $p_i \geq 1$, and $u_{i+1} \cdots u_j \neq 1$ if $c_i = c_j$), $F(c_1), \cdots, F(c_k)$ are fingerprints of $c_1, \ldots, c_k$, then $F(c) = u_1 F(c_1) u_2 F(c_2) \cdots u_k F(c_k) u_{k+1}$.*

**Definition 6.3** *Let $\mathcal{A} = (Q, A, \delta, i, F)$ be a trim deterministic finite state automaton recognizing a regular language $L = L(\mathcal{A})$. Let $q \in Q$ and $c_1, \ldots, c_k$ be cycles such that $\delta(q, c_i) = q$, for $i \in \{1, \ldots, k\}$. We say that $q$ is a multicyclic state in $\mathcal{A}$ if $k > 1$, $i, j \in \{1, \ldots, k\}$, $a, b \in A$ and $a \neq b$ such that $F(c_i) \in aA^*, F(c_j) \in bA^*$. We say that $C_q = \{F(c_1), \ldots, F(c_k)\}$ is the set of the cycles associated with $q$ when $C_q$ is a maximal object in the class of the fingerprints of the cycles $c$ such that $\delta(q, F(c)) = q$ for the order of set inclusion.*

**Remark 6.1** Let $\mathcal{A} = (Q, A, \delta, i, F)$ be a trim deterministic finite state automaton recognizing a regular language $L = L(\mathcal{A})$. Let $q$ be a multicyclic state in $\mathcal{A}$ and let $C_q = \{F(c_1), \ldots, F(c_k)\}$ be the set of the cycles associated with $q$. Suppose that $q' \in Q$ and $u, v \in A^*$ exist such that $\delta(q', uc_{i_1}c_{i_2} \ldots c_{i_j}v) = q'$, where $i_t \in \{1, \ldots, k\}$ and the internal states crossed by the transition are different from $q'$. Then, if $uc_{i_1}c_{i_2} \ldots c_{i_j}v$ is a cycle we must have $i_t \neq i_{t'}$.

**Definition 6.4 (Product of cycles)** *Let $\mathcal{A} = (Q, A, \delta, i, F)$ be a trim deterministic finite state automaton recognizing a regular language $L = L(\mathcal{A})$. Let $q$ be a multicyclic state in $\mathcal{A}$ and let $C_q = \{F(c_1), \ldots, F(c_k)\}$ be the set of the cycles associated with $q$. The set $P_q = \{F(c_{i_1})F(c_{i_2}) \ldots F(c_{i_j}) \mid i_t \in \{1, \ldots, k\},\ i_t \neq i_{t'}\}$ is the set of the products associated with $q$.*

**Lemma 6.1** *Let $\mathcal{A} = (Q, A, \delta, i, F)$ be a trim deterministic finite state automaton recognizing a regular language $L = L(\mathcal{A})$, which is closed under conjugacy relation. Let $\mathcal{C}(L) = \bigcup_{c\ cycle} F(c)$ be the set of the fingerprints $F(c)$ for $L$ associated with $c$, for every cycle $c$ in $\mathcal{A}$. Thus, $\mathcal{C}$ =$C(L)$ is finite.*
*Consider the sets $I_1$ and $R$ defined as follows.*

- Definition of $I_1$. $I_1 = \{w \in A^* \mid \delta(i, w) \in F$ *and* $[\forall x, z \in A^*, c \in A^+,\ q \in Q$ *if* $w = xc^kz$ *with* $\delta(i, x) = q,\ \delta(q, c) = q$, *then* $c = F(c)$ *is a cycle and* $k = 1]\}$.
- Definition of $R$. $R = \bigcup_{c\ cycle} R_c \cup \bigcup_{q\ multicyclic\ state} R_q$, *where $R_c$ and $R_q$ are defined as follows. Since $\mathcal{A}$ is trim and $L$ is closed under the conjugacy relation, then $A^*F(c) \cap L \neq \emptyset$. For each $c$ let $\gamma_{F(c)}$ be a fixed word in $A^*$ such that $\gamma_{F(c)}F(c) \in I_1$. Then, we set*

$$R_c = \{(F(c), \gamma_{F(c)}F(c); F(c)), (\gamma_{F(c)}F(c), F(c); F(c))\}$$

$$R_q = \{(F(c)p, \gamma_{F(c)}F(c); F(c)p), (\gamma_{F(c)}F(c), F(c)p\,; F(c)) \mid F(c) \in C_q,\ p \in P_q,\ F(c)p \in P_q\}$$

*Thus, $I_1$ and $R$ also finite.*

We now define the above-mentioned Pixton system, constructed starting with $L$ and we prove that each word in $^\sim L$ can be generated by this system (Proposition 6.2). At the end of this section we will give an example of a Pixton system constructed by means of definition below.

**Definition 6.5** *Let $\mathcal{A} = (Q, A, \delta, i, F)$ be a trim deterministic finite state automaton recognizing a regular language $L = L(\mathcal{A})$. Let us consider $SC_{PI} = (A, I, R)$, where $I =^\sim I_1$ and $R = \bigcup_{c\ cycle} R_c \cup \bigcup_{q\ multicyclic\ state} R_q$, $I_1$ and $R_c$ defined as in Lemma 6.1.*

**Proposition 6.2** *Let $\mathcal{A} = (Q, A, \delta, i, F)$ be a trim deterministic finite state automaton recognizing a regular language $L = L(\mathcal{A})$ which is closed under conjugacy relation. Let us consider the Pixton system $SC_{PI} = (A, I, R)$ given in Definition 6.5. Thus, $^\sim L \subseteq C = C(SC_{PI})$.*

In Definition 6.6, we introduce a class $\mathcal{F}$ of regular languages which are closed under the conjugacy relation and satisfying the "consistence condition" with respect to the Pixton system $SC_{PI}$ given in Definition 6.5. Proposition 6.3 shows that each language in $\mathcal{F}$ has a circularization which is generated by this Pixton system.

**Definition 6.6 (Consistence condition)** *Let $L = L(\mathcal{A})$ be a regular language, recognized by the trim deterministic finite state automaton $\mathcal{A}$. Suppose that $L$ is closed under conjugacy relation and let $SC_{PI} = (A, I, R)$ be the Pixton system given in Definition 6.5.*
*$L \in \mathcal{F}$ if and only if*

- *for every $R_c = (F(c), \gamma_{F(c)}F(c); F(c)), (\gamma_{F(c)}F(c), F(c); F(c)) \in R$, for every $\sim\epsilon, \sim\epsilon' \in A^*$, if $\sim F(c)\epsilon$, $\sim\gamma_{F(c)}F(c)\epsilon' \in \sim L$ then $\sim\epsilon F(c)\epsilon' F(c) \in \sim L$;*
- *for every $R_q \in R$, for every $F(c) \in C_q$, $p \in P_q$, $F(c)p \in P_q$ such that $(F(c)p, \gamma_{F(c)}F(c); F(c)p)$, $(\gamma_{F(c)}F(c), F(c)p\ ; F(c)) \in R_q$, for every $\sim\epsilon, \sim\epsilon' \in A^*$, if $\sim F(c)p\epsilon$, $\sim\gamma_{F(c)}F(c)\epsilon' \in \sim L$ then $\sim\epsilon F(c)p\epsilon' F(c) \in \sim L$.*

Loosely speaking we force the consistence of the system (i.e. $C(SC_{PI}) = C \subseteq \sim L$) through the consistence condition. In this way, if two words contain sites of a rule then the spliced word belongs to $\sim L$.

**Proposition 6.3** *Let $L$ be a regular language such that $L \in \mathcal{F}$, let $SC_{PI} = (A, I, R)$ be the Pixton system given in Definition 6.5. Then, $C(SC_{PI}) = C \subseteq \sim L$.*

In Proposition 6.6 we also show that the class of the free monoids $X^*$ generated by a regular group code $X$ is a subclass of $\mathcal{F}$. In order to prove this result, we must define the class of languages satisfying an aperiodicity condition. This condition allows us to associate with any closed path $\pi$ in $\mathcal{A}$ a label $c$ of a crossing of $\pi$ such that for each word $w = ucv \in L(\mathcal{A}) = L$, having $c$ as factor and for each $k > 0$, all words $w' = uc^k v$ are in $L(\mathcal{A}) = L$.

Given a trim deterministic finite state automaton $\mathcal{A}$ recognizing $L$, we notice that, for every cycle $c$ of $\mathcal{A}$, there exist $u, v, \in A^*$ such that $uc^*v \subseteq L$. Thus, we give the following definition.

**Definition 6.7 (Aperiodicity condition)** *Let $L$ be a regular language closed under conjugacy relation and $\mathcal{A}$ a trim deterministic finite state automaton recognizing $L$, i.e. $L = L(\mathcal{A})$. Then $L$ satisfies the aperiodicity condition with respect to $\mathcal{A}$ if for each cycle $c$ of $\mathcal{A}$, for every $u', v' \in A^*$, we have that*

$$u'cv' \in L \text{ if and only if } u'c^+v' \subseteq L$$

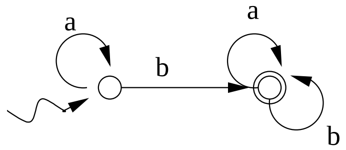


Figure 4: Automaton for $(a^*ba^*)^*$

**Example 6.2** Let us consider the star language $L = (a^*ba^*)^*$ whose deterministic finite state automaton is depicted in Figure 4. We have two cycles $c_1 = a$, $c_2 = b$. We have already observed that $\sim L$ is not a $C_{PA}$ generated and we know that $L$ is not a fingerprint closed star language

[4]. We now prove that $L$ satisfies Definition 6.7. Let us take $c_1 = a$. Since $L$ is the set of words in $A^*$ with at least one $b$, we have that $u'av' \in L$ if and only if $u'a^n v' \in L$, for each $n > 0$. The same argument may be used for the other cycle $b$. Finally, looking at the automaton depicted in Figure 4, we see that the aperiodicity condition is satisfied.

Proposition 6.4 allows us to compare the two conditions in Definitions 6.6 and 6.7.

**Proposition 6.4** *Each language $L$ in $\mathcal{F}$ satisfies the aperiodicity condition.*

The aperiodicity condition given above seems to be related to the definition of *aperiodic languages* [18]. $L \subset A^*$ is *aperiodic* (or noncounting) if and only if there exists $n \in \boldsymbol{N}$ such that for any $u, v, x \in A^*, ux^n v \equiv_L ux^{n+k}v$, for any $k \geq 0$. The least integer $n$ such that the above equivalence holds is called the *index* of $L$ denoted $i(L)$. A well known theorem of Schützenberger states that $L \subseteq A^*$ is aperiodic if and only if $L$ is star free, where $L \subseteq A^*$ is *star free* if $L$ can be obtained from subsets of $A$ by a finite number of set products, union and complements. For example, $A^* = \neg\emptyset$, $(ab)^* = A^* \setminus (bA^* \cup A^*a \cup A^*aaA^* \cup A^*bbA^*)$ are star free languages.

It is quite interesting to investigate the relation between star free languages and languages satisfying the aperiodicity condition. This will be partially done in Proposition 6.5 and Remark 6.2. Proposition 6.5 proves that some star free languages satisfy the aperiodicity condition.

**Proposition 6.5** *Let $L$ be a star free language with $i(L) = 1$. Then $L$ satisfies the aperiodicity condition.*

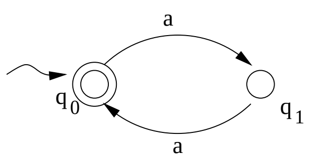


Figure 5: Automaton for $(aa)^*$

**Remark 6.2** Languages exist which are not star free but which satisfies the aperiodicity condition. As an example, let us consider the (star) language $L = (aa)^* \subseteq (a^2)^*$ recognized by the deterministic finite state automaton $\mathcal{A}$ depicted in Figure 5. In [15], the authors observed that $(aa)^*$ is not a star free language, but we now see that $(aa)^*$ satifies the aperiodicity condition. Indeed, we have only one cycle $c = aa$ in $\mathcal{A}$. Furthermore, for every $u', v' \in a^*$ we have

$$\begin{aligned} u'aav' \in L &\Leftrightarrow \exists m \geq 0 \, |u'aav'| = 2m \\ &\Leftrightarrow \exists k \geq 0 \, |u'v'| = 2k \\ &\Leftrightarrow u'aa(aa)^*v' \subseteq L \end{aligned}$$

In Remark 6.3, we give some observations concerning relationship between star free languages and languages whose circularization is $C_{PA}$ or $C_{PI}$ generated.

**Remark 6.3** The class of languages whose circularization is $C_{PA}$ generated cannot be compared with star free languages. Indeed, $(aa)^*$ is not star free, but $\sim(aa)^*$ is $C_{PA}$ generated, while $(a^*ba^*)^*$ is star free, but, as already said, $\sim(a^*ba^*)^*$ is not $C_{PA}$ generated. On the contrary, we do not know whether the class of languages whose circularization is $C_{PI}$ generated can be compared with star free languages. Indeed, we will see that the the star free (star) language $A^* \setminus a^* = (a^*ba^*)^* = A^*bA^*$ belongs to the family $\mathcal{F}$ as well as $(aa)^*$ (which is not star free). Thus, we have that there exist star languages which are not star free and whose circularizations are $C_{PI}$ generated.

In the next proposition we compare star languages generated by regular group codes and star languages satisfying the aperiodicity condition and, by using this result, we state that star languages generated by regular group codes belong to $\mathcal{F}$.

**Proposition 6.6** *Let $X^*$ be a free monoid generated by a regular group code $X$. Then*

1. *$X^*$ satisfies the aperiodicity condition.*
2. *$X^* \in \mathcal{F}$*

The class of languages $X^*$ which are free monoids with $X$ being a regular group code has already been proved to be $C_{PI}$ generated in [4]. Nevertheless, Proposition 6.7 shows that $\mathcal{F}$ also contains other languages.

**Proposition 6.7** *$L = (a^*ba^*)^*$ is a star language which is not generated by a regular group code and which belongs to $\mathcal{F}$.*

**Example 6.3** In the proof of Proposition 6.2 we have given the construction of a Pixton splicing system generating a language in $\mathcal{F}$. We now illustrate this construction for $L = (a^*ba^*)^*$. Looking at the automaton depicted in Figure 4, we see that $\sim L$ is $C_{PI}$ generated by the Pixton system $SC_{PI} = (A, I, R)$, where $I = \sim\{b, ab, ba, bb\}$ and $R = \{(a, ba; a), (ba, a; a), (b, bb; b), (bb, b; b)\} \cup \{(ab, ba; ab), (ba, ab; a), (ba, bb; ba), (bb, ba; b)\}$.

# References


[1] J. Berstel, D. Perrin (1985), "Theory of codes", Academic Press, New York.

[2] J. Berstel, A. Restivo (1981), Codes et sousmonoides fermes par conjugaison, *Sem. LITP* **81-45**, 10 pages.

[3] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza (2001), DNA and circular splicing, *in* "DNA 2000", Lecture Notes in Computer Science **2054**, pp. 117 – 129.

[4] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza (2001), Circular splicing and regularity, submitted.

[5] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza (2001), On the power of circular splicing, *in preparation.*

[6] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza (1999), Linear and circular splicing, *WORDS99.*

[7] C. De Felice (1996), An application of Hajós factorizations to variable-length codes, *Theoretical Computer Science* **164**, 223 – 252.

[8] L. Fuchs (1960), "Abelian groups", Pergamon Press, Oxford, London, New York and Paris.

[9] T. Head (1987), Formal Language Theory and DNA: an analysis of the generative capacity of specific recombinant behaviours, *Bull. Math. Biol.* **49**, 737 – 759.

[10] T. Head (1999), Circular suggestions for DNA Computing, *in* "Pattern Formation in Biology, Vision and Dynamics", (A. Carbone, M. Gromov, P. (B. Pruzinkiewicz, Eds.), World Scientific, Singapore and London, to appear.

[11] T. Head, Gh. Paun, D. Pixton (1996), Language theory and molecular genetics: generative mechanisms suggested by DNA recombination, *in* "Handbook of Formal Languages" (G. Rozenberg & A. Salomaa, Eds.), Vol. 2, pp. 295 – 360, Springer Verlag.

[12] T. Head, G. Rozenberg, R. Bladergroen, C. Breek, P. Lommerse, H. Spaink (2000), Computing with DNA by operating on plasmids, *BioSystems* **57**, 87 – 93.

[13] J.E. Hopcroft, J.D. Ullman (1979), "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley, Reading, Mass.

[14] M. Lothaire (1983), "Combinatorics on Words", Encyclopedia of Math. and its Appl., Addison Wesley Publishing Company.

[15] R. McNaughton, S. Papert (1971), "Counter-Free Automata", MIT Press, Cambridge, Mass. .

[16] G. Paun (1996), On the splicing operation, *Discrete Applied Math.* **70**, 57 – 79.

[17] G. Paun, G. Rozenberg, A. Salomaa (1998), "DNA computing, New Computing Paradigms", Springer-Verlag.

[18] D. Perrin, Finite Automata, *in* "Handbook of Theoretical Computer Science" (J. Van Leeuwen, Ed.), Vol. $B$, pp. 1 – 57, Elsevier.

[19] D. Pixton (1997), Splicing in abstract families of languages, *Theoret. Comp. Science* to appear.

[20] D. Pixton (1996), Linear and Circular Splicing Systems, *in* "Proc. of 1st Int. Symp. on Int. in Neural and Biological Systems", (May 1996) 181 – 188.

[21] D. Pixton (1996), Regularity of splicing languages, *Discrete Applied Math.* **69**, 101 – 124.

[22] C. Reis, G. Thierren (1979), Reflective star languages and codes, *Information and Control* **42**, 1 – 9.

[23] R. Siromoney, K.G. Subramanian, A. Dare (1992), Circular DNA and Splicing Systems, *in* "Proc. of ICPIA", Lecture Notes in Computer Science **654**, Springer-Verlag, pp. 260 – 273.