

## ON BONDED SEQUENTIAL AND PARALLEL INSERTION SYSTEMS<sup>☆</sup>

MARKUS HOLZER<sup>1,\*</sup>, BIANCA TRUTHE<sup>1</sup> AND AHMAD FIRDAUS YOSMAN<sup>2</sup>

**Abstract.** We introduce a new variant of insertion systems, namely bonded insertion systems. In such systems, words are not only formed by usual letters but also by bonds between letters. Words which can be inserted, have “free” bonds at their ends which control at which positions in a word they can be inserted (namely only there, where the bonds “fit”). Two kinds of bonded insertion systems are defined in this paper: so-called bonded sequential insertion systems and bonded parallel insertion systems. In a sequential system, there is only one word inserted at a time. In a parallel system, there is a word inserted at every possible position in parallel in one time step. We investigate the generative capacity of those two kinds and relate the families of generated languages to some families of the Chomsky hierarchy and to families of languages generated by Lindenmayer systems. Additionally, we investigate some closure properties.

**Mathematics Subject Classification.** 68Q42, 68Q45.

Accepted November 21, 2018.

### 1. INTRODUCTION

The operations of (*sequential*) *insertion* and *deletion*, defined and studied in [8] are generalizations of the operations of concatenation and left/right quotients. We remark that the operation of sequential insertion was already investigated in [3]. Intuitively, for given two strings  $u$  and  $v$ , instead of concatenating  $v$  at the right end of  $u$ , the insertion operation interpolates  $v$  in an arbitrary place in  $u$ . Conversely, the deletion operation, instead of removing  $v$  from the left or right ends of  $u$ , extracts  $v$  from an arbitrary position in  $u$ .

Several variants of these operations have been studied in the literature. In [8], the variants of parallel, controlled and scattered insertion and deletion were introduced and investigated. Contextual variants were introduced and studied first in [19]. For further reading regarding these variants, we refer to [11–13, 16, 18]. Necessary and sufficient conditions, under which the result of sequential insertion or deletion of two strings is a singleton set, are given in [9]. In [10], a natural generalisation of both left and right derivative operations, called derivative, has been considered, where a substring can be deleted from an arbitrary position in a string. The closure, decidability and other properties of language families under derivatives have been investigated. The

---

<sup>☆</sup>This is an extended version of an article presented at NCMA 2016 [4]. Some of the results presented in this paper were obtained during the research stay of the last author at the Institut für Informatik of Universität Giessen, Germany.

*Keywords and phrases:* Bonded insertion systems, sequential insertion, parallel insertion, formal languages, generative power.

<sup>1</sup> Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Germany.

<sup>2</sup> Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia.

\* Corresponding author: [holzer@informatik.uni-giessen.de](mailto:holzer@informatik.uni-giessen.de)

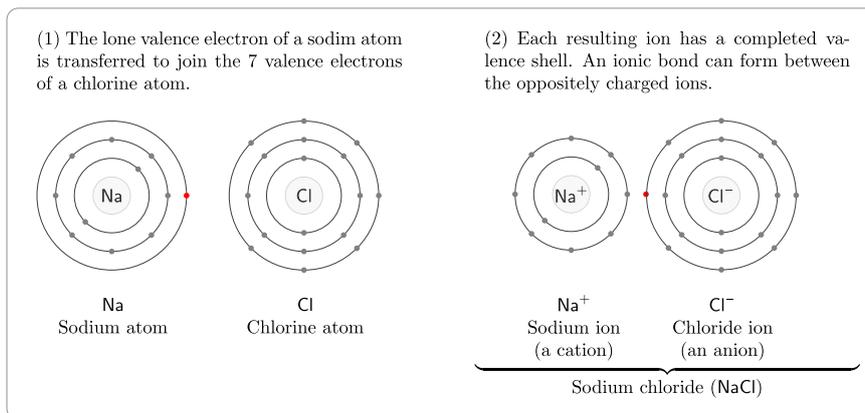


FIGURE 1. Ionic bonding.

problems related to the cardinality, effective construction and decidability of deletion sets, which are the sets of substrings obtained by erasing them from a string have been discussed in [14]. Similar issues of the sets arising from a given string by ( $k$ -bounded) parallel deletions of substrings belonging to a given language have been considered in [15]. Computability and decidability problems of contextual insertion and deletion operations have been investigated in [19]. Insertion, shuffle, deletion and scattered deletion closures of languages have been investigated in [6, 7]. Block insertions and deletions on trajectories, which correspond to spaces at the beginning, at the end and between two symbols, have been studied in [1].

Though insertion and deletion operations are originally motivated from linguistics, they have recently become of wide interest in relation to recombinant phenomena of DNA and RNA molecules in genetics. An *insertion-deletion* system (an *insdel system*, for short), a powerful theoretical model of DNA computing, is based on contextual insertion and deletion operations, which serve as formal models of insertion and deletion processes of DNA strands by mismatching annealing. The fundamental results on insdel systems can be found in [2, 16, 21] whereas the recent results have appeared in [17, 20].

In this paper, we show that insertion-deletion operations and systems can describe “atomic” level behaviours of bio-molecular compounds: we introduce a new variant of insertion operations on strings and languages, called *bonded sequential and parallel insertions*, as formal models of (bio-)chemical reactions that make and break chemical bonds. Let us consider the ionic bonding process of a sodium atom (Na) and a chlorine atom (Cl) resulting in the ionic compound sodium chloride (NaCl). Here, Na has 11 electrons with 1 electron in its valence (last) shell and Cl has 17 electrons with 7 electrons in its valence shell. When these atoms encounter each other, the valence electron of Na is transferred to Cl yielding complete valence shells. As a result, Na has 11 protons and 10 electrons giving it a net electrical charge of  $1+$ , Cl has 17 protons and 18 electrons giving it net electrical charge of  $1-$ . Thus, the ratio of  $\text{Na}^+$  and  $\text{Cl}^-$  is  $1 : 1$ , *i.e.*, “balanced”—see, Figure 1.

The other chemical bonds, such as covalent bonds, hydrogen bonds, have also similar balanced ratios (see [22]). The process of (bio-)chemical reactions that make and break chemical bonds produce (large) molecules with stabilized (balanced) ratios, allowing to define insertion-deletion systems in a (bio-)molecular framework.

Motivated by this framework, we consider strings over a “bonding” alphabet, *i.e.*, an alphabet of symbols with left non-negative and right non-positive bonds. A string  $u$  over some bonding alphabet is said to be *well-formed* if the sum of the right and left bond at each joining of two symbols is zero. If, additionally, the sum of the left bond of the first symbol and the right bond of the last symbol is also zero,  $u$  is called *balanced*, otherwise it is called *unbalanced*. For two strings  $u$  and  $v$  over a bonding alphabet, we impose the condition that the insertion of  $v$  into  $u$  can be performed if the strings  $u$  and  $v$  are balanced and the obtained string is well-formed (and hence, also balanced).

The paper is organized as follows: in the next section, we introduce the necessary notations on finite automata, context-free grammars and Lindenmayer systems. In Section 3, we define bonding alphabets and bonding insertion systems. In Section 4, we prove relations to Chomsky language families (for sequential systems) and to Lindenmayer systems (for parallel systems). In Section 5, we investigate some closure properties. In Section 6, we summarize our results, give an open problem and mention further research directions.

## 2. PRELIMINARIES

We assume that the reader is familiar with the basic concepts of formal language theory (see, *e.g.*, [24]). We here only recall some notations used in the paper.

The cardinality of a set  $S$  is denoted by  $|S|$ ; the set of all finite subsets of a set  $S$  is denoted by  $2^S$ . The inclusion of a set  $A$  in a set  $B$  is denoted by  $A \subseteq B$  and the proper inclusion by  $A \subset B$ .

Let  $\Sigma$  be an alphabet. By  $\Sigma^*$  we denote the set of all words over the alphabet  $\Sigma$ . A language  $L$  over an alphabet  $\Sigma$ , for short  $L \subseteq \Sigma^*$ , is a set of words. The empty word is denoted by  $\lambda$ . For a word  $w$ , we denote the length by  $|w|$ .

A (*nondeterministic*) *finite automaton* (NFA) is a 5-tuple  $A = (Q, \Sigma, \delta, q_0, F)$ , where  $Q$  is a finite set of *states*,  $\Sigma$  is a finite set of *input symbols*,  $q_0 \in Q$  is called the *initial state*,  $F \subseteq Q$  is a set of *accepting states* and  $\delta : Q \times \Sigma \rightarrow 2^Q$  is a *transition function*. The *language accepted* by an NFA  $A$  is defined as

$$L(A) = \{w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset\},$$

where the transition function  $\delta$  is recursively extended to  $\delta^* : Q \times \Sigma^* \rightarrow 2^Q$  in the natural way by

$$\delta^*(p, \lambda) = \{p\} \quad \text{and} \quad \delta^*(p, aw) = \bigcup_{q \in \delta(p, a)} \delta^*(q, w).$$

If there is no danger of confusion, we simply write  $\delta$  instead of  $\delta^*$ . Moreover, a finite automaton is *deterministic* (a DFA) if and only if for all  $q \in Q$  and  $a \in \Sigma$  the set  $\delta(q, a)$  is a singleton, *i.e.*,  $|\delta(q, a)| = 1$ . It is well known that DFAs or NFAs characterize the family of regular languages  $\mathcal{L}(\text{REG})$ .

A *context-free grammar* is a 4-tuple  $G = (N, T, P, S)$  where  $N$  and  $T$  are disjoint alphabets of nonterminals and terminals, respectively,  $S \in N$  is the *starting nonterminal*, and  $P$  is a finite set of *productions* of the form  $A \rightarrow \alpha$  where  $A \in N$  and  $\alpha \in (N \cup T)^*$ .

The derivation relation  $\Rightarrow_G$  of a context-free grammar  $G$  is defined as follows:  $\alpha \Rightarrow_G \beta$  if and only if  $\alpha = \alpha_1 A \alpha_2$ , for some words  $\alpha_1, \alpha_2 \in (N \cup T)^*$  and a nonterminal  $A \in N$  and there is a production  $A \rightarrow \gamma$  in  $P$  and  $\beta = \alpha_1 \gamma \alpha_2$ . The reflexive and transitive closure of  $\Rightarrow_G$  is denoted by  $\Rightarrow_G^*$ . If there is no danger of confusion, we write  $\Rightarrow$  and  $\Rightarrow^*$  instead of  $\Rightarrow_G$  and  $\Rightarrow_G^*$ , respectively.

The *language generated* by a context-free grammar  $G = (N, T, P, S)$  is the set

$$L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}.$$

The family of all context-free languages is referred to as  $\mathcal{L}(\text{CF})$ .

We give now some definitions regarding Lindenmayer systems (or L-systems for short). A *TOL-system* is a triple  $G = (\Sigma, H, \omega)$ , where  $\Sigma$  denotes a finite alphabet,  $H$  is a finite set of finite substitutions  $\Sigma_i : \Sigma \rightarrow 2^{\Sigma^*}$  with  $1 \leq i \leq n$  for some natural number  $n$  and  $\omega \in \Sigma^*$  is called the *axiom*. Instead of  $w \in \Sigma_i(a)$  for  $1 \leq i \leq n$  and  $a \in \Sigma$ , we write  $a \rightarrow w$ . A word  $u \in \Sigma^*$  is derived to a word  $v \in \Sigma^*$ , written as  $u \Rightarrow_G v$ , if and only if there is a substitution  $\Sigma \in H$  such that  $v \in \Sigma(u)$ . The reflexive and transitive closure of  $\Rightarrow_G$  is denoted by  $\Rightarrow_G^*$ . If there is no danger of confusion, we write  $\Rightarrow$  and  $\Rightarrow^*$  instead of  $\Rightarrow_G$  and  $\Rightarrow_G^*$ , respectively. The *language generated* by  $G$  is defined as

$$L(G) = \{x \mid \omega \Rightarrow^* x\}.$$

An *extended* TOL-system (an ETOL-system) is a 4-tuple  $G = (\Sigma, H, \omega, \Delta)$ , where  $G' = (\Sigma, H, \omega)$  is a TOL-system (the underlying TOL-system) and  $\Delta \subseteq \Sigma$ . The *language generated* by an ETOL-system  $G$  is

$$L(G) = L(G') \cap \Delta^*,$$

where  $G'$  is the underlying TOL-system of  $G$ .

A TOL-system  $G = (\Sigma, H, \omega)$  is said to be a DTOL-system if  $H$  contains only homomorphisms. Moreover, if in a TOL- or DTOL-system the set  $H$  is a singleton set, *i.e.*, contains only one finite substitution or one homomorphism, then we speak of a 0L- or D0L-system, respectively. These notations also apply to extended systems like EDTOL-, E0L- and ED0L-systems.

The family of all languages generated by a variant

$$X \in \{\text{ETOL}, \text{E0L}, \text{EDTOL}, \text{ED0L}, \text{TOL}, \text{0L}, \text{DTOL}, \text{D0L}\}$$

of Lindenmayer systems is denoted by  $\mathcal{L}(X)$ .

### 3. BONDED SYSTEMS

We introduce the basic structure where the bonded insertion systems are working on, namely the bonding alphabet. To this end, let  $\mathbb{Z}$  be the set of integers as well as

$$\mathbb{Z}_0^- = \{0, -1, -2, \dots\} \quad \text{and} \quad \mathbb{Z}_0^+ = \{0, 1, 2, \dots\}.$$

Observe, that  $\mathbb{Z}_0^+ = \mathbb{N}$ .

Let  $\Sigma$  be an alphabet. Then the set  $\mathcal{B}_\Sigma = \mathbb{Z}_0^+ \times \Sigma \times \mathbb{Z}_0^-$  is a *bonding alphabet* over  $\Sigma$ . An element  $(i, a, -j)$  of  $\mathcal{B}_\Sigma$  is called a *letter*  $a$  with left bond  $i$  and right bond  $-j$ . To simplify the presentation, we write  $[i \ a \ -j]$  instead of  $(i, a, -j)$  for a letter from  $\mathcal{B}_\Sigma$ . Let

$$w = [i_0 \ a_1 \ i_1][i_2 \ a_2 \ i_3][i_4 \ a_3 \ i_5] \cdots [i_{2n-2} \ a_n \ i_{2n-1}]$$

be a non-empty sequence of letters from  $\mathcal{B}_\Sigma$ . The sequence  $w$  is said to be *well-formed* if all the bonds fit, *i.e.*, if  $i_{2j-1} + i_{2j} = 0$  for  $1 \leq j \leq n-1$ . If additionally  $i_0 + i_{2n-1} = 0$  holds, then  $w$  is said to be a *balanced word* or for short a *word*. In case  $i_0 + i_{2n-1} \neq 0$ , then the word is said to be *unbalanced*. Moreover, a word is *neutral* if  $i_0 = i_{2n-1} = 0$ . For a well-formed word

$$w = [i_0 \ a_1 \ -i_1][i_1 \ a_2 \ -i_2][i_2 \ a_3 \ -i_3] \cdots [i_{n-1} \ a_n \ -i_n],$$

we say that the word  $w$  has the left bond  $i_0$  and the right bond  $-i_n$  as the *outer bonds* and  $i_1, \dots, i_{n-1}$  as *inner bonds*. If we are not interested in the inner bonds, we shortly write

$$[i_0 \ a_1 a_2 a_3 \cdots a_n \ -i_n].$$

The set of all well-formed words built from letters of  $\mathcal{B}_\Sigma$  including the empty word is referred to as  $\mathcal{B}_\Sigma^*$  and the set of all balanced words built by letters from  $\mathcal{B}_\Sigma$  including the empty word is referred to as  $\mathcal{B}_\Sigma^\circledast$ . By definition  $\mathcal{B}_\Sigma^\circledast \subset \mathcal{B}_\Sigma^*$ . The empty word is the neutral element of both structures  $\mathcal{B}_\Sigma^*$  and  $\mathcal{B}_\Sigma^\circledast$ . For the *empty word*, we write  $[i_0 \ \lambda \ -i_0]$  for some number  $i_0 \in \mathbb{Z}_0^+$ . The empty word is always a balanced word. The length of a bond word  $w$  from  $\mathcal{B}_\Sigma^*$  or  $\mathcal{B}_\Sigma^\circledast$  is denoted by  $|w|$  and is equal to the number of letters in  $w$ . In particular, the empty bond word is of length 0.

Analogously to ordinary insertion systems, we define bonded sequential insertion systems as follows:

**Definition 3.1.** A *bonded sequential insertion system* (a bSINS-system for short) is a triple  $\gamma = (\Sigma, A, I)$ , where  $\Sigma$  is a finite alphabet,  $A \subseteq \mathcal{B}_\Sigma^{\otimes}$  is a finite set of *axioms* that contains only neutral words and  $I \subseteq \mathcal{B}_\Sigma^{\otimes}$  is a finite set of *insertion strings*.

The derivation relation  $\Rightarrow$  of a bSINS-system  $\gamma = (\Sigma, A, I)$  is defined in the following way: let  $\alpha, \beta \in \mathcal{B}_\Sigma^{\otimes}$ . Then  $\alpha \Rightarrow_\gamma \beta$  if and only if  $\alpha = \alpha_1 \alpha_2$ , for two words  $\alpha_1, \alpha_2 \in \mathcal{B}_\Sigma^*$  and there is an insertion string  $\alpha' \in I$  such that  $\beta = \alpha_1 \alpha' \alpha_2$ . Observe that, by definition, concatenation to the left and right is also possible. The reflexive and transitive closure of  $\Rightarrow_\gamma$  is denoted by  $\Rightarrow_\gamma^*$ . If there is no danger of confusion, we write  $\Rightarrow$  and  $\Rightarrow^*$  instead of  $\Rightarrow_\gamma$  and  $\Rightarrow_\gamma^*$ , respectively.

We define a homomorphism

$$h_{\text{be}} : \mathcal{B}_\Sigma^{\otimes} \rightarrow \Sigma^* \quad \text{by} \quad h_{\text{be}}([i \ a \ -j]) = a \quad \text{for every } [i \ a \ -j] \in \mathcal{B}_\Sigma$$

and call it *bond erasing homomorphism*. The *language generated* by a bSINS-system  $\gamma = (\Sigma, A, I)$  is defined as

$$L(\gamma) = \{h_{\text{be}}(\beta) \mid \text{there is an axiom } \alpha \in A \text{ such that } \alpha \Rightarrow_\gamma^* \beta\}.$$

The family of all such languages is denoted by  $\mathcal{L}(\text{bSINS})$ .

In order to clarify our notation, we give an example.

**Example 3.2.** Let  $\Sigma = \{a, b\}$ . We define a bSINS-system  $\gamma = (\Sigma, A, I)$ , where the set of axioms contains the neutral words

$$[0 \ \lambda \ 0], \quad [0 \ b \ 0], \quad [0 \ a \ -1][1 \ b \ 0], \quad \text{and} \quad [0 \ b \ -1][1 \ b \ 0],$$

and the set of insertion strings contains the balanced words

$$[1 \ a \ -1] \quad \text{and} \quad [1 \ b \ -1].$$

Consider the example derivation

$$\begin{aligned} [0 \ a \ -1][1 \ b \ 0] &\Rightarrow [0 \ a \ -1][1 \ a \ -1][1 \ b \ 0] \\ &\Rightarrow [0 \ a \ -1][1 \ b \ -1][1 \ a \ -1][1 \ b \ 0] \\ &\Rightarrow [0 \ a \ -1][1 \ b \ -1][1 \ a \ -1][1 \ a \ -1][1 \ b \ 0], \end{aligned}$$

which gives the word *abaab* by applying the bond erasing homomorphism  $h_{\text{be}}$  described above.

By construction, the two words from  $I$  can be inserted at an arbitrary inner position of any word derived from the axiom  $[0 \ a \ -1][1 \ b \ 0]$  or  $[0 \ b \ -1][1 \ b \ 0]$ . Thus, by starting from the former axiom, any derived string (after bond erasing) contains at least one letter  $b$ , while in the latter case the derived word contains at least two letters  $b$ . In any case, the obtained word ends with  $b$ . Hence, we conclude by taking the other axioms into consideration, that  $L(\gamma) \subseteq (a^*b)^*$ . Conversely, we argue as follows: let

$$w = a^{i_1} b a^{i_2} b \dots a^{i_k} b$$

be an arbitrary word in  $(a^*b)^*$ . In order to show that  $w \in L(\gamma)$ , we consider several cases:

1. if  $k = 0$ , then  $w = \lambda$ , which can be produced by using the axiom  $[0 \ \lambda \ 0]$  without any derivation. Thus,  $\lambda \in L(\gamma)$ .

2. Let  $k = 1$ . If  $i_1 = 0$ , then  $w = b$ . This corresponds to the axiom  $[0 \ b \ 0]$  of  $\gamma$ . Therefore, we have  $b \in L(\gamma)$ , too. Now let  $i_1 > 0$ . Then  $w = a^{i_1}b$  and there is a derivation

$$\begin{aligned} [0 \ a \ -1][1 \ b \ -0] &\Rightarrow [0 \ a \ -1][1 \ a \ -1][1 \ b \ -0] \Rightarrow \dots \\ &\Rightarrow [0 \ a \ -1][1 \ a \ -1]^{i_1-1}[1 \ b \ -0] \end{aligned}$$

in  $\gamma$  that inserts letters  $[1 \ a \ -1]$ , which proves the containment  $w \in L(\gamma)$ .

3. Finally, let  $k > 1$ . As above, we consider two sub-cases. Assume  $i_1 = 0$ . Then, we have  $w = ba^{i_2}b \dots a^{i_k}b$  and there is a derivation starting with the axiom  $[0 \ b \ -1][1 \ b \ 0]$  that first inserts  $k - 2$  letters  $[1 \ b \ -1]$  and then inserts  $i_2 + i_3 + \dots + i_k$  letters  $[1 \ a \ -1]$ . Thus, the derivation looks as follows:

$$\begin{aligned} [0 \ b \ -1][1 \ b \ 0] &\Rightarrow \dots \Rightarrow [0 \ b \ -1][1 \ b \ -1]^{k-2}[1 \ b \ 0] \Rightarrow \dots \\ &\Rightarrow [0 \ b \ -1][1 \ a \ -1]^{i_2}[1 \ b \ -1]^{k-2}[1 \ b \ 0] \Rightarrow \dots \\ &\Rightarrow [0 \ b \ -1][1 \ a \ -1]^{i_2}[1 \ b \ -1] \dots [1 \ b \ -1][1 \ a \ -1]^{i_k}[1 \ b \ 0]. \end{aligned}$$

This shows that  $w \in L(\gamma)$ . The last sub-case  $i_1 > 0$  follows with a similar argumentation as above, now starting with the axiom  $[0 \ a \ -1][1 \ b \ 0]$ .

Therefore, the converse inclusion holds, too. Thus  $L(\gamma) = (a^*b)^*$ .

We now define bonded parallel insertion systems where, in contrast to sequential systems, insertions take place at every possible position in one step in parallel. The formal definition of such a parallel system does not differ from the definition of a sequential one, however, the definition of the derivation is different.

**Definition 3.3.** A *bonded parallel insertion system* (a bPINS-system for short) is a triple

$$\gamma = (\Sigma, A, I),$$

where  $\Sigma$  is a finite alphabet,  $A \subseteq \mathcal{B}_\Sigma^\otimes$  is a finite set of *axioms* that contains only neutral words and  $I \subseteq \mathcal{B}_\Sigma^\otimes$  is a finite set of *insertion strings*.

The derivation relation  $\Rightarrow$  of a bPINS-system  $\gamma = (\Sigma, A, I)$  is defined as follows: let words  $\alpha, \beta \in \mathcal{B}_\Sigma^\otimes$ . Then  $\alpha \Rightarrow_\gamma \beta$  if and only if  $\alpha = \alpha_1\alpha_2 \dots \alpha_n$  for non-empty subwords  $\alpha_i \in \mathcal{B}^*(\Sigma)$  with  $1 \leq i \leq n$  and  $\beta = \delta_1\alpha_1\delta_2\alpha_2 \dots \delta_{n-1}\alpha_{n-1}\delta_n\alpha_n\delta_{n+1}$  for insertion strings  $\delta_i \in I$  for  $2 \leq i \leq n$  and  $\delta_1 \in I$  if an insertion is possible at the left end of the word  $\alpha$  or  $\delta_1 = \lambda$  otherwise and  $\delta_{n+1} \in I$  if an insertion is possible at the right end of the word  $\alpha$  or  $\delta_{n+1} = \lambda$  otherwise and there is no insertion string which can be inserted inside some word  $\alpha_i$  with  $1 \leq i \leq n$  (for any decomposition  $\alpha_i = u_i v_i$  with  $1 \leq i \leq n$  and insertion string  $\zeta \in I$ , the bonded word  $u_i \zeta v_i$  is not well-formed). Please note that at bonds between letters, for which no insertion string exists (which could be inserted without destroying that the word is well-formed), no insertion takes place but the derivation continues. At any bond where an insertion string can be inserted, an insertion string is actually inserted.

The reflexive and transitive closure of  $\Rightarrow_\gamma$  is denoted by  $\Rightarrow_\gamma^*$ . If there is no danger of confusion we write  $\Rightarrow$  and  $\Rightarrow^*$  instead of  $\Rightarrow_\gamma$  and  $\Rightarrow_\gamma^*$ , respectively.

The *language generated* by a bPINS-system  $\gamma = (\Sigma, A, I)$  is defined as

$$L(\gamma) = \{h_{\text{be}}(\beta) \mid \text{there is an axiom } \alpha \in A \text{ such that } \alpha \Rightarrow_\gamma^* \beta\},$$

where  $h_{\text{be}}$  is the bond erasing homomorphism defined on Page 131. The family of all such languages is denoted by  $\mathcal{L}(\text{bPINS})$ .

In order to clarify our notation, we give an example.

**Example 3.4.** Let  $\gamma = (\{a\}, A, I)$  be a bPINS-system with

$$A = \{[{}_0 a {}_0], [{}_0 a {}_{-1}][{}_1 a {}_0]\} \quad \text{and} \quad I = \{[{}_1 a {}_{-2}][{}_2 a {}_{-1}]\}.$$

The words of the set  $A$  yield the words  $a$  and  $aa$ . The word  $[{}_1 a {}_{-2}][{}_2 a {}_{-1}]$  has one position where an insertion can be applied. After inserting, a word of length 4 is obtained which has two insertion positions. Inside the insertion word, no insertion is possible. Hence, in each derivation step, from a word of length  $2^n$  for some natural number  $n \geq 1$  (which always has  $2^{n-1}$  possible insertion positions), the word with length  $2^{n+1}$  is obtained which has  $2^n$  positions where an insertion can be applied. Other words are not generated. Thus, the bPINS-system  $\gamma$  generates the language  $\{a^{2^n} \mid n \geq 0\}$ .

#### 4. ON THE GENERATIVE POWER

In the following subsections, we investigate the generative power of bonded insertion systems. We relate the family of languages generated sequentially or parallelly to language families of the Chomsky hierarchy and Lindenmayer hierarchy. Further, we show that the systems form an infinite hierarchy with respect to the number of different bonds occurring a system.

##### 4.1. Bonded sequential insertion systems

We remark that bonded sequential insertion systems can also be expressed as restricted contextual insertion systems where an insertion rule is of a special form (which takes care of the correct form of an insertion word) and which has always one letter as the left context and one letter as the right context (in order to ensure that after an insertion the bonds still fit) and where at the end a homomorphism is applied to obtain the “terminal” words. However, due to these restrictions, the results from [21] cannot be directly taken although they look sometimes similar to our results.

Our first result on bonded sequential insertion systems is that they can generate any regular language.

**Theorem 4.1.** *Let  $A$  be a nondeterministic finite automaton without  $\lambda$ -transitions. Then there is a bonded sequential insertion system  $\gamma$  such that  $L(\gamma) = L(A)$ .*

*Proof.* Let  $A = (Q, \Sigma, \delta, q_0, F)$  be an NFA and set  $L = L(A)$ . Without loss of generality, we may assume that (i) the initial state of  $A$  is not reachable by any state of  $A$  by a non-empty word, *i.e.*, the automaton is non-returning and (ii) that there are at most two accepting states, namely the initial state  $q_0$  if  $\lambda \in L$  and an accepting state that is a dead state, *i.e.*, there are no out-going transitions. By renaming the states, we may further assume that  $Q = \{0, 1, 2, \dots, n-1\}$ , for some  $n \geq 1$ , and that  $q_0 = 0$  and  $F = \{n-1\} \cup \{0 \mid \lambda \in L\}$ .

Before we construct a bonded sequential insertion system  $\gamma = (\Sigma, A, I)$ , we need some notation in order to simplify the presentation. Let  ${}_p A_q = (Q, \Sigma, \delta, p, \{q\})$ . For each word  $w = a_1 a_2 \cdots a_m$  with  $a_i \in \Sigma$ , for  $1 \leq i \leq m$ , we associate a (unbalanced) word w.r.t.  ${}_p A_q$  as follows: if  $p_0, p_1, \dots, p_m$  is a sequence of states which appear in an accepting computation of  ${}_p A_q$  on  $w$ , that is

1.  $p = p_0$ ,
2.  $p_{i+1} \in \delta(p_i, a_{i+1})$ , for  $0 \leq i < m$  and
3.  $p_m = q$ ,

then the associated (unbalanced) word is

$$[{}_{p_0} a_1 {}_{-p_1}][{}_{p_1} a_2 {}_{-p_2}] \cdots [{}_{p_{m-1}} a_m {}_{-p_m}].$$

In case  $p_m = n-1$ , we associate the word

$$[{}_{p_0} a_1 {}_{-p_1}][{}_{p_1} a_2 {}_{-p_2}] \cdots [{}_{p_{m-1}} a_m {}_{-p_0}].$$

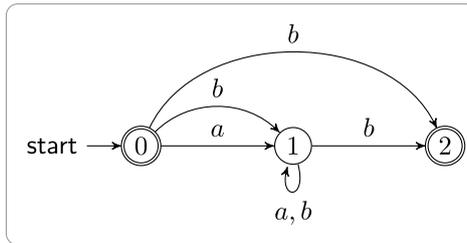


FIGURE 2. Nondeterministic finite automaton  $A$  accepting the language  $(a^*b)^*$ .

instead. Observe, that the only difference between these two words is their rightmost bond value. Now we are ready to describe the bonded sequential insertion system.

Let  $\gamma = (\Sigma, A, I)$ . The set of axioms contains all neutral (balanced) words over the bonding alphabet  $\mathcal{B}_\Sigma$  which are associated with words  $w$  which are accepted by the automaton  $A$ , where in the computation all states are pairwise different (no state appears twice or more often). Observe, that  $[{}_0 \lambda {}_0] \in A$  if  $\lambda \in L$ . Similarly, we define the insertion set  $I$  to contain all balanced words over the bonding alphabet  $\mathcal{B}_\Sigma$  which are associated with words  $w$  accepted by the automaton  ${}_p A_p$ , for  $p \in Q \setminus \{0, n-1\}$ , where in the computation all states are pairwise different, except for the first and last state. Note that all words in  $I$  came from words that were induced by a loop-computation in the automaton  $A$ .

By construction, it is easy to see that  $L(\gamma) \subseteq L(A)$ , since in all words from the axiom and in the insertion, the bonds encode the transition function  $\delta$  of the automaton  $A$ . Moreover, the well-formedness of these words guarantees that the bonds fit, which means that each word encodes a valid computation of  $A$ . The insertion relation  $\Rightarrow_\gamma$  is defined in such a way that well-formedness remains, which means that a sub-computation coming from a loop-computation is inserted. Finally, because the homomorphism  $h_{\text{be}}$  in the definition of  $L(\gamma)$  deletes the bonds, we end up with a word that is accepted by  $A$ . Hence,  $L(\gamma) \subseteq L(A)$ . The converse inclusion  $L(A) \subseteq L(\gamma)$  can be shown by similar arguments and is left to the reader. Thus, the stated claim follows.  $\square$

An example explains the previous construction.

**Example 4.2.** Consider the regular language  $L = (a^*b)^*$  which is accepted by the finite automaton  $A = (Q, \Sigma, \delta, q_0, F)$ , where the set of states is  $Q = \{0, 1, 2\}$ , the input alphabet is  $\Sigma = \{a, b\}$ , the initial state is  $q_0 = 0$ , the set of accepting states is  $F = \{0, 2\}$  and the transition function is given by

$$\delta(q, a) = \{1\} \text{ and } \delta(q, b) = \{1, 2\} \text{ for } q \in \{0, 1\}.$$

The NFA  $A$  is depicted in Figure 2.

It is easy to see that the NFA  $A$  has no  $\lambda$ -transitions and fulfills the conditions on the initial state and on the accepting states used in the proof of Theorem 4.1.

Next, we construct the bonded sequential insertion system  $\gamma = (\Sigma, B, I)$ . Because  $\lambda \in (a^*b)^*$ , the word  $[{}_0 \lambda {}_0]$  belongs to the axiom set  $B$ . From the non-empty words accepted by  $A$ , only  $b$ ,  $ab$  and  $bb$  have computations without repeating states. Since  $b$  is accepted *via* the state sequence 0, 2, the word  $[{}_0 b {}_0]$  is associated with it. Since  $ab$  is accepted *via* the state sequence 0, 1, 2, we associate the word  $[{}_0 a {}_{-1}] [{}_1 b {}_0]$  with it. For  $bb$ , it is the word  $[{}_0 b {}_{-1}] [{}_1 b {}_0]$ . Thus, the axiom set  $B$  consists of the words

$$[{}_0 \lambda {}_0], \quad [{}_0 b {}_0], \quad [{}_0 a {}_{-1}] [{}_1 b {}_0], \quad \text{and} \quad [{}_0 b {}_{-1}] [{}_1 b {}_0].$$

Next, we have a look on the words for the insertion set. The only words that are accepted by  ${}_1 A_1$  without repeating states are  $a$  and  $b$ . To these two words, we associate

$$[{}_1 a {}_{-1}] \quad \text{and} \quad [{}_1 b {}_{-1}]$$

over  $\mathcal{B}_\Sigma^\otimes$ . Therefore, the set is  $I = \{[{}_1 a \_1], [{}_1 b \_1]\}$ . In Example 3.2, we have shown that exactly this bonded sequential insertion system  $\gamma$  generates the language  $(a^*b)^*$  which is stated also by Theorem 4.1.

Since, for any nondeterministic finite automaton  $A$ , there is a bonded sequential insertion system which generates the language accepted by the automaton  $A$  (Thm. 4.1), we conclude:

**Corollary 4.3.**  $\mathcal{L}(\text{REG}) \subset \mathcal{L}(\text{bSINS})$ .

*Proof.* The inclusion follows from Theorem 4.1. For the strictness we argue as follows: consider the bonded sequential insertion system  $\gamma = (\Sigma, A, I)$  with alphabet

$$\Sigma = \{a, b\},$$

axiom set

$$A = \{[{}_0 \lambda \_0], [{}_0 a \_0][{}_0 b \_0]\},$$

and insertion set

$$I = \{[{}_0 a \_0][{}_0 b \_0]\}.$$

It is easy to see that  $L(\gamma)$  is the Dyck language on one parenthesis pair  $a$  and  $b$  (including the empty word). Thus, we obtain  $L(\gamma) \cap a^*b^* = \{a^n b^n \mid n \geq 0\}$ , which is a non-regular language. Thus, the strict inclusion follows.  $\square$

In our next theorem, we give a relation to controlled sequential insertion systems introduced by Kari [8] which we recall here. A *controlled sequential insertion system* is a triple  $\gamma = (\Sigma, A, \Delta)$ , where  $\Sigma$  is a finite alphabet,  $A \subseteq \Sigma^*$  is a finite set of *axioms* and  $\Delta : \Sigma \rightarrow \Sigma^*$  is a *finite substitution*. The derivation  $\Rightarrow_\gamma$  is defined as follows:  $\alpha \Rightarrow_\gamma \beta$  if and only if  $\alpha = \alpha_1 a \alpha_2$ , for  $\alpha_1, \alpha_2 \in \Sigma^*$  and  $a \in \Sigma$  and  $\beta = \alpha_1 a v_a \alpha_2$ , where  $v_a \in \Delta(a)$ . As usual, the reflexive and transitive closure of  $\Rightarrow_\gamma$  is denoted by  $\Rightarrow_\gamma^*$ . If there is no danger of confusion, we write  $\Rightarrow$  and  $\Rightarrow^*$  instead of  $\Rightarrow_\gamma$  and  $\Rightarrow_\gamma^*$ , respectively.

The *language generated* by a controlled sequential insertion system  $\gamma = (\Sigma, A, \Delta)$  is defined as

$$L(\gamma) = \{w \in \Sigma^* \mid \text{there is an axiom } \alpha \in A \text{ such that } \alpha \Rightarrow_\gamma^* w\}.$$

The next theorem relates a special form of bonded sequential insertion systems and controlled sequential insertion systems, showing that they are almost the same.

**Theorem 4.4.** *Let  $\gamma = (\Sigma, A, I)$  be a bonded sequential insertion system, where the insertion set  $I$  does not contain any neutral words. Then there is a controlled sequential insertion system  $\gamma'$  and a homomorphism  $h$  such that  $L(\gamma) = h(L(\gamma'))$ .*

*Proof.* Let  $\gamma = (\Sigma, A, I)$  be a bonded sequential insertion system with the above mentioned property. Then the position of any insertion that can be performed by the system  $\gamma$  must lie within a word. Thus, appending words at the beginning or at the end of the word is *not* possible.

Define the maximal bond value  $\ell$  of  $\gamma$  as

$$\ell = \max\{|i|, |j| \mid \text{there is } [{}_i a \_j] \in \mathcal{B}_\Sigma \text{ appearing in a word in } A \text{ or } I\}.$$

We further need a homomorphism  $g : \mathcal{B}_\Sigma^\otimes \rightarrow (\Sigma')^*$  defined by  $g([{}_i a \_j]) = (a, j)$ , for letters  $[{}_i a \_j] \in \mathcal{B}_\Sigma$ ,

Then we define a controlled sequential insertion system  $\gamma' = (\Sigma', A', \Delta)$  where the alphabet is  $\Sigma' = \Sigma \times \{0, 1, \dots, \ell\}$ , the set of axioms is  $A' = \{g(w) \mid w \in A\}$  and the substitution is defined as

$$\Delta((a, j)) = \{g(w) \mid w \in I \text{ and the left bond of } w \text{ is } j\}$$

for  $(a, j) \in \Sigma'$ . Because  $A$  and  $I$  are finite sets, the axiom set  $A'$  is finite and  $\Delta$  is a finite substitution, too.

By construction, one observes that whenever there is a derivation step in  $\gamma$  of the form  $\alpha \Rightarrow_{\gamma} \beta$  with  $\alpha = \alpha_1 \alpha_2$ , for  $\alpha_1, \alpha_2 \in \mathcal{B}_{\Sigma}^*$  and there is an insertion string  $\alpha' \in I$  such that  $\beta = \alpha_1 \alpha' \alpha_2$ , then it can be mimicked in  $\gamma'$  by the derivation

$$g(\alpha) \Rightarrow_{\gamma'} g(\beta)$$

with  $g(\alpha) = g(\alpha_1 \alpha_2) = g(\alpha_1)g(\alpha_2)$  and  $g(\alpha_1)$  ends with a letter  $(a, j)$ , where  $j$  is the left bond value of  $\alpha_2$  and  $g(\alpha') \in \Delta((a, j))$ , such that

$$g(\alpha_1)g(\alpha')g(\alpha_2) = g(\alpha_1 \alpha' \alpha_2) = g(\beta)$$

and *vice versa*. Here it is essential that the derivation in  $\gamma$  cannot append a word to the left. Due to the restriction that  $\gamma$  has no neutral words in the insertion set, no derivation in  $\gamma$  concatenates to the left. Hence, in both systems we essentially find the same derivations. Therefore, we can conclude that  $L(\gamma) = h(L(\gamma'))$ , where

$$h : (\Sigma')^* \rightarrow \Sigma^*$$

is defined by  $h((a, j)) = a$  for  $(a, j) \in \Sigma'$ . □

Next we show that the property on bonded sequential insertion systems required in the previous theorem is not a restriction at all. In fact, the property that the insertion set does not contain any neutral words can be seen as a *normalform* for bonded sequential insertion systems.

**Theorem 4.5.** *Let  $\gamma = (\Sigma, A, I)$  be a bonded sequential insertion system. Then there is a bonded sequential insertion system  $\gamma' = (\Sigma, A', I')$ , where the insertion set  $I'$  does not contain any neutral words, such that  $L(\gamma') = L(\gamma)$ .*

*Proof.* Let  $k$  be a bond that is not present in  $\gamma$ . Then one replaces the neutral words of the insertion set by words where all zero bond values are replaced by a bond value that is not present in the original system. To this end let  $\text{all} : \mathcal{B}_{\Sigma}^{\otimes} \rightarrow \mathcal{B}_{\Sigma}^{\otimes}$  be defined by replacing all zero bond values occurring as a left bond by  $k$  and as a right bond by  $-k$  such that the word stays balanced. For instance,

$$\text{all}([{}_0 a \ -1][{}_1 b \ -2][{}_2 c \ 0]) = [{}_k a \ -1][{}_1 b \ -2][{}_2 c \ -k]$$

and

$$\text{all}([{}_0 a \ -1][{}_1 b \ 0][{}_0 c \ -2][{}_2 d \ 0]) = [{}_k a \ -1][{}_1 b \ -k][{}_k c \ -2][{}_2 d \ -k].$$

Obviously, this mapping generalizes to set of words. Then, the insertion  $I'$  of  $\gamma'$  can be described as  $I' = \text{all}(I)$ .

In order to simulate a derivation in the original system, we now have to replace inner zero bond values (these are bond values that are neither on the left or on the right of a word) in the words from the axiom set with  $-k$  and  $k$  such that the words stay balanced. Moreover, we have to take care on insertions to the left and right applied to the axioms, with the neutral words from  $I$ . Let

$$I_0 = \{\alpha \mid \alpha \text{ is a neutral word in } I\}.$$

Therefore, we define

$$A' = \text{inner}(A) \cup \text{inner}(I_0 \cdot A) \cup \text{inner}(A \cdot I_0) \cup \text{inner}(I_0 \cdot A \cdot I_0),$$

where  $\text{inner} : \mathcal{B}_\Sigma^{\otimes} \rightarrow \mathcal{B}_\Sigma^{\otimes}$  is defined by replacing all inner zero bond values by  $-k$  and  $k$ , respectively, such that the word stays balanced. For instance,

$$\text{inner}([{}_0 a \ -_1][{}_1 b \ -_2][{}_2 c \ 0]) = [{}_0 a \ -_1][{}_1 b \ -_2][{}_2 c \ 0]$$

and

$$\text{inner}([{}_0 a \ -_1][{}_1 b \ -_0][{}_0 c \ -_2][{}_2 d \ 0]) = [{}_0 a \ -_1][{}_1 b \ -_k][{}_k c \ -_2][{}_2 d \ 0].$$

As in the case of the mapping  $\text{all}$ , also the mapping  $\text{inner}$  generalizes to sets of words. This completes the description of the bSINS-system  $\gamma'$ . Then it is easy to see, that  $L(\gamma') = L(\gamma)$ . The details are left to the reader.  $\square$

We continue with an upper bound on the generative capacity of bonded sequential insertion systems.

**Theorem 4.6.** *Let  $\gamma$  be a bonded sequential insertion system. Then there is a context-free grammar  $G$  such that  $L(G) = L(\gamma)$ .*

*Proof.* The idea for the simulation of a bonded sequential insertion system by a context-free grammar is to represent the bonds by nonterminal symbols. For each possible insertion point, we use a nonterminal that allows one to insert a string from the insertion set of  $\gamma$ . More formally, the construction reads as follows.

Let  $\gamma = (\Sigma, A, I)$  be a bSINS-system. Define the maximal bond value  $\ell$  of  $\gamma$  as

$$\ell = \max\{|i|, |j| \mid \text{there is a letter } [{}_i a \ -_j] \in \mathcal{B}_\Sigma \text{ that appears in a word in } A \text{ or } I\}.$$

Then we construct a context-free grammar  $G = (N, T, P, S)$  where the set  $N$  of nonterminals is defined by (the union being disjoint)

$$\begin{aligned} N = & \{[X_i, a] \mid 0 \leq i \leq \ell \text{ and } a \in \Sigma\} \\ & \cup \{[X_i, a, X_j] \mid 0 \leq i, j \leq \ell \text{ and } a \in \Sigma \cup \{\lambda\}\} \\ & \cup \{S\}, \end{aligned}$$

the set of terminals is  $T = \Sigma$  and the production set  $P$  contains the following rules: the derivation starts with a rule of the form

$$\begin{aligned} S & \rightarrow [X_0, a, X_0] && \text{if } a \in \Sigma \cup \{\lambda\} \text{ and } [{}_0 a \ 0] \in A, \\ S & \rightarrow g(u)[X_i, a, X_0] && \text{if } a \in \Sigma, u \in \mathcal{B}_\Sigma^* \text{ and } u[{}_i a \ 0] \in A, \end{aligned}$$

where  $g : \mathcal{B}_\Sigma^* \rightarrow N^*$  is defined via  $g([{}_i a \ -_j]) = [X_i, a]$ . Then the derivation continues by simulating an insertion. This is done by the rules

$$\begin{aligned} [X_i, a] & \rightarrow [X_i, a] && \text{if } [{}_i \lambda \ -_i] \in I, \\ [X_i, a] & \rightarrow [X_i, b][X_i, a] && \text{if } b \in \Sigma \text{ and } [{}_i b \ -_i] \in I, \\ [X_i, a] & \rightarrow g([{}_i b \ -_j]u[{}_k c \ -_i])[X_i, a] && \text{if } [{}_i b \ -_j]u[{}_k c \ -_i] \in I \text{ with} \\ & && [{}_i b \ -_j], [{}_k c \ -_i] \in \mathcal{B}_\Sigma \text{ and } u \in \mathcal{B}_\Sigma^* \end{aligned}$$

and

$$\begin{array}{ll}
[X_i, a, X_0] & \rightarrow [X_i, a, X_0] & \text{if } a \in \Sigma \text{ and } [{}_i \lambda \text{ } _i] \in I, \\
[X_i, a, X_0] & \rightarrow [X_i, b][X_i, a, X_0] & \text{if } \{a, b\} \subseteq \Sigma \text{ and } [{}_i b \text{ } _i] \in I, \\
[X_i, a, X_0] & \rightarrow g([{}_i b \text{ } _j]u[{}_k c \text{ } _i])[X_i, a, X_0] & \text{if } a \in \Sigma \text{ and } [{}_i b \text{ } _j]u[{}_k c \text{ } _i] \in I \text{ with} \\
& & \quad [{}_i b \text{ } _j], [{}_k c \text{ } _i] \in \mathcal{B}_\Sigma \text{ and } u \in \mathcal{B}_\Sigma^*, \\
[X_i, a, X_0] & \rightarrow [X_i, a, X_0] & \text{if } a \in \Sigma \text{ and } [{}_0 \lambda \text{ } _0] \in I, \\
[X_i, a, X_0] & \rightarrow [X_i, a][X_0, b, X_0] & \text{if } \{a, b\} \subseteq \Sigma \text{ and } [{}_0 b \text{ } _0] \in I, \\
[X_i, a, X_0] & \rightarrow [X_i, a]g([{}_0 b \text{ } _j]u)[X_k, c, X_0] & \text{if } a \in \Sigma \text{ and } [{}_0 b \text{ } _j]u[{}_k c \text{ } _0] \in I \text{ with} \\
& & \quad [{}_0 b \text{ } _j], [{}_k c \text{ } _0] \in \mathcal{B}_\Sigma \text{ and } u \in \mathcal{B}_\Sigma^*,
\end{array}$$

and

$$\begin{array}{ll}
[X_0, \lambda, X_0] & \rightarrow [X_0, \lambda, X_0] & \text{if } [{}_0 \lambda \text{ } _0] \in I, \\
[X_0, \lambda, X_0] & \rightarrow [X_0, b, X_0] & \text{if } b \in \Sigma \text{ and } [{}_0 b \text{ } _0] \in I, \\
[X_0, \lambda, X_0] & \rightarrow g([{}_0 b \text{ } _j]u)[X_k, c, X_0] & \text{if } [{}_0 b \text{ } _j]u[{}_k c \text{ } _0] \in I \text{ with} \\
& & \quad [{}_0 b \text{ } _j], [{}_k c \text{ } _0] \in \mathcal{B}_\Sigma \text{ and } u \in \mathcal{B}_\Sigma^*.
\end{array}$$

Finally, the termination is done by

$$[X_i, a] \rightarrow a \quad \text{and} \quad [X_i, a, X_j] \rightarrow a,$$

for  $a \in \Sigma$  or  $a \in \Sigma \cup \{\lambda\}$ , respectively and  $0 \leq i, j \leq \ell$ . This completes the description of the context-free grammar  $G$ .

The idea behind the nonterminals is that any bond word from  $\mathcal{B}_\Sigma^*$  (restricted to bond values used in  $\gamma$ ) can be encoded by a string of nonterminals. For instance, the bond word  $[{}_0 a \text{ } _1][{}_1 b \text{ } _2][{}_2 c \text{ } _0]$  will be encoded as  $[X_0, a][X_1, b][X_2, c, X_0]$ . Then by construction it is not hard to see that any insertion of a word from  $I$  into the bond word can be simulated by an application of an appropriate rule described above at the correct corresponding position within the sentential form derived by the grammar. For instance, if in the bond word  $[{}_0 a \text{ } _1][{}_1 b \text{ } _2][{}_2 c \text{ } _0]$  we want to insert  $[{}_1 c \text{ } _3][{}_3 d \text{ } _1]$  leading to  $[{}_0 a \text{ } _1][{}_1 c \text{ } _3][{}_3 d \text{ } _1][{}_1 b \text{ } _2][{}_2 c \text{ } _0]$  this can be simulated by the grammar as follows:

$$[X_0, a][X_1, b][X_2, c, X_0] \Rightarrow [X_0, a][X_1, c][X_3, d][X_1, b][X_2, c, X_0].$$

If the insertion in  $\gamma$  takes place at the rightmost position, one has to be careful, since the encoding of such a string enforces that the last symbol is a triple. Nevertheless, an inspection of the presented rules above, shows that this issue is already taken care. The tedious details of the straightforward proof showing that  $L(G) = L(\gamma)$  is left to the reader.  $\square$

As a corollary, we obtain the strict inclusion of  $\mathcal{L}(\text{bSINS})$  within  $\mathcal{L}(\text{CF})$ .

**Corollary 4.7.** *The strict inclusion  $\mathcal{L}(\text{bSINS}) \subset \mathcal{L}(\text{CF})$  holds.*

*Proof.* The inclusion follows from Theorem 4.6. For the strictness we show that one cannot generate the context-free language

$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

by any bonded sequential insertion system. Here  $w^R$  refers to the mirror image of  $w$ .

Assume to the contrary that there is a bonded insertion system  $\gamma = (\{a, b\}, A, I)$  with  $L(\gamma) = L$ . Define

$$\ell = \max\{|w| \mid w \in A \cup I\}.$$

Then we consider any derivation in  $\gamma$  that leads to a word  $\alpha$  whose image under the bond erasing homomorphism is  $a^n b b a^n$ , for some  $n > 3\ell$ . That is,  $h(\alpha) = a^n b b a^n$ , where  $h$  is the bond erasing homomorphism. By the choice of  $n$ , one deduces that in any derivation that we consider at least 2 insertion steps inserting non-empty words from  $I$  happened. Since not all insertion steps in our derivation can only insert words whose image under the bond erasing homomorphism only contains of  $a$ 's we may safely assume that one of the above insertion steps introduces both  $b$ 's in one step, while the other derivation step only introduces a *non-zero* number of letters  $a$  (all with appropriate bond values), say the balanced word  $\alpha'$ . The order in which these two derivation steps appear in the whole derivation of the word  $\alpha$  is *not* relevant. What is important is the fact, that the point of insertion of  $\alpha'$  is still present in  $\alpha$ . It is easy to see that the point of insertion lies either in the left half or in the right half of  $\alpha$ —it cannot lie inside the subword  $[{}_i b \text{ } -_j] [{}_j b \text{ } -_k]$  of  $\alpha$ , since  $|\alpha'| \geq 1$ . In either case, inserting  $\alpha'$  again, produces a word whose image under the bond erasing homomorphism is not of the form  $w w^R$ , for any  $w \in \{a, b\}^*$ . If  $\alpha'$  is inserted within the left half of  $\alpha$ , the subword  $[{}_i b \text{ } -_j] [{}_j b \text{ } -_k]$  which marks the border between the word and its mirror image is shifted to the right and thus the derived word is not of the appropriate form anymore. A similar argumentation applies if the insertion happens in the right half of  $\alpha$ . This is a contradiction to our assumption and, therefore, the language  $L$  is not a member of  $\mathcal{L}(\text{bSINS})$ .  $\square$

The inclusion  $\mathcal{L}(\text{bSINS}) \subset \mathcal{L}(\text{CF})$  shows even more, namely that every language generated by a bonded sequential insertion system has a semi-linear Parikh image. This is due to the fact that every context-free language has a semi-linear Parikh image.

For an alphabet  $\Sigma = \{a_1, a_2, \dots, a_k\}$  with a fixed order of the letters, the *Parikh-mapping* is a mapping  $\psi : \Sigma^* \rightarrow \mathbb{N}^k$  defined by

$$\psi(w) = (|w|_{a_1}, |w|_{a_2}, \dots, |w|_{a_k}).$$

This mapping is extended to languages  $L \subseteq \Sigma^*$  by  $\psi(L) = \{\psi(w) \mid w \in L\}$ . A language  $L$  is called *semi-linear* if its Parikh-image is a semi-linear set. For vectors  $\vec{x}_0, \vec{x}_1, \dots, \vec{x}_n \in \mathbb{N}^k$ , the set

$$\{\vec{x}_0 + \ell_1 \cdot \vec{x}_1 + \ell_2 \cdot \vec{x}_2 + \dots + \ell_n \cdot \vec{x}_n \mid \ell_i \in \mathbb{N}, \text{ for } 1 \leq i \leq n\}$$

is *linear*. A subset of  $\mathbb{N}^k$  is said to be *semi-linear* if it is a finite union of linear subsets.

The importance of semi-linear sets stems from the fact that any unary context-free language is already regular. This is a simple application of the semi-linearity of context-free languages. Thus, we can state the next theorem on unary languages generated by bonded sequential insertion systems.

**Corollary 4.8.** *Every unary language in  $\mathcal{L}(\text{bSINS})$  is regular.*

For any natural number  $n \geq 1$ , let  $\mathcal{L}(\text{bSINS}_n)$  denote the family of all languages which are generated by a bSINS-system where at most  $n$  different, non-negative bonds occur in the words (axioms and insertion strings). These families form an infinite hierarchy.

**Theorem 4.9.** *For any natural number  $n \geq 1$ , we have the strict inclusion*

$$\mathcal{L}(\text{bSINS}_n) \subset \mathcal{L}(\text{bSINS}_{n+1}).$$

*Proof.* Any bSINS-system with  $n$  bonds can be simulated by a system with  $n + 1$  bonds (for instance, by one with one more insertion string which has new outer bonds and will therefore never be used). We now show the properness of this inclusion.

Let  $n \geq 1$  and  $\Sigma_n = \{a_1, \dots, a_n\}$  be a set of  $n$  pairwise distinct letters. Further, let

$$L_n = \{a_1^{p_1} a_2^{p_2} \dots a_n^{p_n} a_{n+1} \mid p_i \geq 1 \text{ and } 1 \leq i \leq n\}.$$

This language, as can easily be seen, is generated by the bSINS-system

$$\gamma_n = (\Sigma_n, \{[0 \ a_1 \ -1][1 \ a_2 \ -2] \cdots [n-1 \ a_n \ -n][n \ a_{n+1} \ 0]\}, \{[i \ a_i \ -i] \mid 1 \leq i \leq n\})$$

which has  $n + 1$  non-negative bonds. Hence,  $L_n \in \mathcal{L}(\text{bSINS}_{n+1})$ .

Assume, the language  $L_n$  also belongs to the family  $\mathcal{L}(\text{bSINS}_n)$ . Then, there is a bSINS-system

$$\gamma'_n = (\Sigma_n, A_n, I_n)$$

which only has  $n$  non-negative bonds but also generates the language  $L_n$ . Let

$$\ell = \max\{|w| \mid w \in A_n\}.$$

Consider a word

$$[i_0 \ a_1^{\ell_1} \ -i_1][i_1 \ a_1^{\ell'_1} a_2^{\ell_2} \ -i_2] \cdots [i_{n-1} \ a_{n-1}^{\ell'_{n-1}} a_n^{\ell_n} \ -i_n][i_n \ a_n^{\ell'_n} a_{n+1} \ i_0]$$

with  $\ell_j + \ell'_j = \ell$  and bonds  $i_j \geq 0$  for  $0 \leq j \leq n$  which is generated by the system  $\gamma'_n$ . Due to the choice of  $\ell$ , this word is not an axiom and, moreover, a copy of each letter had been inserted in order to obtain this word. Let the numbers  $\ell_j$  with  $0 \leq j \leq n$  be chosen in such a way that at least one copy of each letter  $a_j$  had been inserted at the bond with  $i_j$ . Since  $L_n \in \mathcal{L}(\text{bSINS}_n)$  (our assumption), there are at least two equal bonds  $i_r$  and  $i_s$  with  $0 \leq r < s \leq n$ . But then, in the next derivation step, a copy of the letter  $a_r$  can be inserted at the bond  $i_s$  and, hence, behind the letter  $a_s$ , which yields a word which has not the required structure. This contradiction proves that  $n$  bonds are not sufficient. Thus,  $L_n \notin \mathcal{L}(\text{bSINS}_{n+1}) \setminus \mathcal{L}(\text{bSINS}_n)$   $\square$

Since the witness languages used in the proof of the previous theorem are all regular, we immediately see that the family of the regular languages is not included in any of the families  $\mathcal{L}(\text{bSINS}_n)$ . On the other hand, we know from the proof of Corollary 4.3 that the Dyck language over one parenthesis pair (which is not regular) belongs to the family  $\mathcal{L}(\text{bSINS}_1)$ .

**Corollary 4.10.** *The family  $\mathcal{L}(\text{REG})$  is incomparable to any family  $\mathcal{L}(\text{bSINS}_n)$  for  $n \geq 1$ .*

## 4.2. Bonded parallel insertion systems

From Example 3.4, we obtain our first result on bonded parallel insertion systems.

**Lemma 4.11.** *The relation  $\{a^{2^n} \mid n \geq 0\} \in \mathcal{L}(\text{bPINS}) \setminus \mathcal{L}(\text{bSINS})$  holds.*

*Proof.* Let  $L = \{a^{2^n} \mid n \geq 0\}$ . This language is generated by the bPINS-system  $\gamma$  of Example 3.4. By Corollary 4.8, we know that every unary language generated by a bSINS-system is regular. Since  $L$  is unary but not regular, the language cannot be generated by a bSINS-system.  $\square$

Our next result on bonded parallel insertion systems is that they can simulate any bonded sequential insertion system.

**Theorem 4.12.** *Let  $\gamma$  be a bonded sequential insertion system. Then there is a bonded parallel insertion system  $\gamma'$  such that  $L(\gamma') = L(\gamma)$ .*

*Proof.* Let  $\gamma = (\Sigma, A, I)$  be a bonded sequential insertion system. We construct a bonded parallel insertion system  $\gamma' = (\Sigma, A, I')$  as follows.

As insertion words of the set  $I'$ , we take every insertion rule of the set  $I$  and additionally the empty insertion words  $[i \ \lambda \ -i]$  for every bond  $(-i, i)$  which occurs as an insertion position in an axiom or in an insertion word of the set  $I$  (for which there are a left bond  $j$ , a right bond  $k$  and two letters  $x$  and  $y$  such that  $[j \ x \ -i][i \ y \ -k]$  is a subword of an axiom or an insertion word of the set  $I$ ).

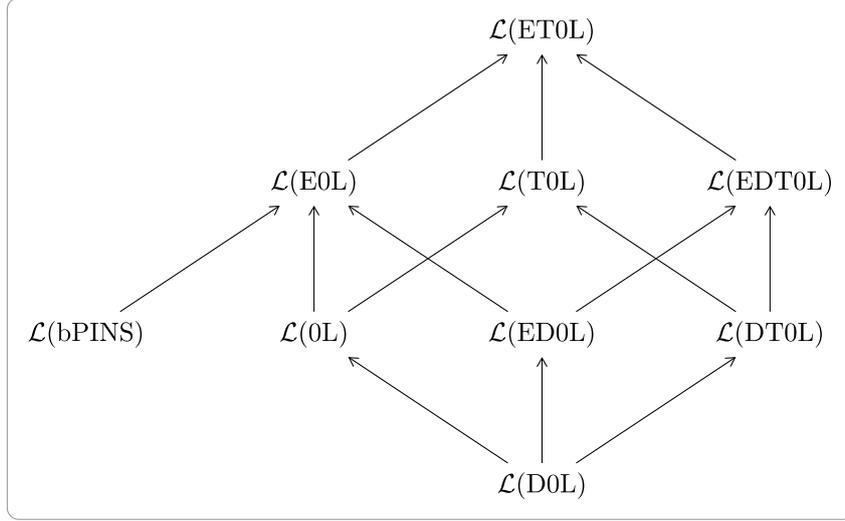


FIGURE 3. Relation to the hierarchy of L-systems. An arrow from an entry  $X$  to an entry  $Y$  denotes the proper inclusion  $X \subset Y$ . If two entries  $X$  and  $Y$  are not connected by a directed path, then the classes  $X$  and  $Y$  are incomparable.

Any derivation of the bSINS-system  $\gamma$  can be simulated by the bPINS-system  $\gamma'$  by starting from the same axiom and applying, in each step, the same insertion at the same insertion position as in the sequential system and inserting suitable empty words at every other position. Hence,  $L(\gamma) \subseteq L(\gamma')$ . On the other hand, for any derivation of the bPINS-system  $\gamma'$ , there is also a derivation of the bSINS-system  $\gamma$  to the same word which starts with the same axiom and applies sequentially every insertion which is applied in parallel in one step by the parallel system  $\gamma'$  (sequential insertions of the words of the set  $I' \setminus I$  do not change the word and can therefore be skipped). Hence,  $L(\gamma') \subseteq L(\gamma)$  and together, we obtain  $L(\gamma') = L(\gamma)$ .  $\square$

The parallel insertion of words cannot necessarily be simulated sequentially (because, in this mode, “intermediate” words are generated which might not be elements of the language generated by the parallel system). A language which can be generated parallelly but not sequentially was given in Lemma 4.11.

The previous statements together yield the following result.

**Corollary 4.13.** *The proper inclusion  $\mathcal{L}(\text{bSINS}) \subset \mathcal{L}(\text{bPINS})$  holds.*

*Proof.* The inclusion  $\mathcal{L}(\text{bSINS}) \subseteq \mathcal{L}(\text{bPINS})$  follows from Theorem 4.12. The properness follows from Lemma 4.11.  $\square$

With respect to Lindenmayer systems, we have the relations depicted in Figure 3. In order to prove the relations given in Figure 3, we first present a language which is generated by an L-system but not by an bPINS-system.

**Lemma 4.14.** *Let  $L = \{a^{2^n} b^{n-1} c \mid n \geq 1\}$ . Then  $L \in \mathcal{L}(\text{D0L}) \setminus \mathcal{L}(\text{bPINS})$ .*

*Proof.* Let  $\Gamma = (\{a, b, c\}, \{a \rightarrow aa, b \rightarrow b, c \rightarrow bc\}, aac)$  be a D0L-system. The axiom  $aac$  belongs to the language  $L$ . A word  $a^{2^n} b^{n-1} c \in L$  is derived by the system  $\Gamma$  in one step to the word  $a^{2^{n+1}} b^{n-1} bc$  which also belongs to the language  $L$ . Hence,  $L(\Gamma) \subseteq L$ . On the other hand, any word  $a^{2^n} b^{n-1} c \in L$  is obtained in  $n - 1$  derivation steps from the axiom  $aac$ . Hence,  $L \subseteq L(\Gamma)$ . Thus,  $L \in \mathcal{L}(\text{D0L})$ .

We now prove that the language  $L$  cannot be generated by a bPINS-system.

Assume that the language is generated by a bPINS-system  $\gamma = (\{a, b, c\}, A, I)$ . Let

$$\begin{aligned}\ell_A &= \max\{|w| \mid w \in A\}, \\ \ell_I &= \max\{|w| \mid w \in I\},\end{aligned}$$

and

$$\ell_\gamma = \ell_A + \ell_I + 1.$$

We consider the word

$$w = a^{2^{\ell_\gamma}} b^{\ell_\gamma - 1} c \in L.$$

Due to the choice of  $\ell_\gamma$ , this word is obtained from a shorter word by inserting at least one non-empty insertion string  $u$ . If there is an insertion rule applicable which contains different basic letters, then deriving the bond word  $b_w$  belonging to  $w$  would yield a word with more than one subword  $ab$  or  $bc$  (because every non-empty applicable insertion string is applicable at least twice in the next step). Hence, every insertion string has either only letters  $a$  or only letters  $b$  (every word of the language contains exactly one  $c$ , hence, every applicable insertion string does not contain the letter  $c$ ). Since with the insertion of the word  $u$  only one kind of letters has been inserted (only letters  $a$  or letters  $b$ ), there is also an insertion word  $v$  with the other kind of letters which has been inserted in order to obtain the bond word  $b_w$ . Without loss of generality, let us assume that the word  $v$  consists of letters  $b$ . After the insertion of the word  $v$ , the derived word has at least twice as many insertion positions where the word  $v$  can be inserted (at both ends of the previously inserted words). In every derivation step, this number will at least be doubled. Hence, there is a bond word  $b_{w'}$  derived which has at least  $\ell_I + 2$  insertion positions where the word  $v$  can be inserted. Let the number of letters  $b$  in this word be denoted by  $k$ . Then the number of letters  $a$  is  $2^{k+1}$ . In the next step, at every considered insertion position, the word  $v$  can be inserted. This yields a word with  $l_b$  letters  $b$  and

$$l_b \geq k + (\ell_I + 2) \cdot |v| \geq k + \ell_I + 2.$$

The number  $l_a$  of letters  $a$  in this word is  $l_a = 2^{l_b+1}$ . Hence,

$$l_a \geq 2^{k+3} 2^{\ell_I}.$$

On the other hand, one can insert into the bond word  $b_{w'}$  at most  $(2^{k+1} + 1) \cdot \ell_I$  new letters  $a$ . Thus,

$$l_a \leq 2^{k+1} + (2^{k+1} + 1) \cdot \ell_I \leq 2^{k+1} + 2^{k+2} \ell_I \leq 2^{k+3} \ell_I < 2^{k+3} 2^{\ell_I} \leq l_a.$$

This contradiction shows that the language  $L$  cannot be generated by a bPINS-system. □

We now prove the relations stated in Figure 3.

**Theorem 4.15.** *The relations depicted in Figure 3 hold.*

*Proof.* The hierarchy of the language classes defined by the variants of Lindenmayer systems is well known in the theory of L systems (see [23]).

We now prove

1.  $\mathcal{L}(\text{bPINS}) \subset \mathcal{L}(\text{EOL})$  and
2.  $\mathcal{L}(\text{bPINS})$  is incomparable to the classes  $\mathcal{L}(\text{T0L})$ ,  $\mathcal{L}(\text{DT0L})$ ,  $\mathcal{L}(\text{0L})$ ,  $\mathcal{L}(\text{D0L})$ ,  $\mathcal{L}(\text{EDT0L})$  and  $\mathcal{L}(\text{ED0L})$ .

We first prove that the inclusion  $\mathcal{L}(\text{bPINS}) \subseteq \mathcal{L}(\text{EOL})$  holds.

Let  $\gamma = (\Sigma, A, I)$  be a bPINS-system. We show how to construct an EOL-system which generates the same language.

The idea for the simulation of a parallel insertion system by an E0L-system is basically the same as for the simulation of a sequential system by a context-free grammar. We represent a well-formed word of the set  $\mathcal{B}_\Sigma^*$  by a sequence of symbols where each but the last symbol indicates a letter of  $\Sigma$  together with its left bond and the last letter indicates a letter of  $\Sigma$  together with both its bonds. Note that the rightmost bond of any axiom and bond word derived from it is zero.

For each bond  $b$  occurring in an axiom or an insertion string, let  $I_b$  be the set of all insertion strings which have  $b$  as their left bonds:  $I_b = \{[b w -b] \mid [b w -b] \in I\}$ . For a set  $M$  of bond words, let  $\text{alph}(M)$  be the set of all letters of the bonding alphabet which occur in a word of the set  $M$  and let  $\text{suf}(M)$  be the set of all letters of the bonding alphabet which occur at the end of a word of the set  $M$ . Further, we define a set  $N_1$  of symbols corresponding to letters in an axiom or insertion string together with their left bonds,

$$N_1 = \{[X_i, a] \mid [i a -j] \in \text{alph}(A \cup I) \text{ for some } j \in \mathbb{Z}_0^+\},$$

and a set  $N_2$  of symbols corresponding to the last letters in an axiom or neutral insertion string together with both their bonds,

$$N_2 = \{[X_i, a, X_0] \mid [i a 0] \in \text{suf}(A \cup I)\}.$$

We now construct an E0L-system  $\Gamma = (N \cup \Sigma, \Sigma, S, \Sigma)$  which simulates the derivation of the bPINS-system  $\gamma$ . We define the set  $N$  of new symbols as follows:

$$N = N_1 \cup N_2 \cup \{S, F\}.$$

Let  $\mathcal{N}_\Sigma^\otimes$  be the set of all neutral balanced words over  $\Sigma$ . Let  $f : \mathcal{B}_\Sigma^\otimes \rightarrow N^*$  and  $g : \mathcal{N}_\Sigma^\otimes \rightarrow N^*$  be mappings which transform a bond word into representations for the E0L-system. They are defined as follows: an empty bond word  $[i \lambda -i]$  is mapped to the empty word  $\lambda$  by both mappings. A non-empty bond word

$$w = [i_1 a_1 -i_2][i_2 a_2 -i_3] \cdots [i_n a_n -i_{n+1}]$$

with  $a_i \in \Sigma$  for  $1 \leq i \leq n$  is mapped by  $f$  to the word

$$f(w) = [X_{i_1}, a_1][X_{i_2}, a_2] \cdots [X_{i_n}, a_n].$$

If the word  $w$  is neutral ( $i_1 = i_{n+1} = 0$ ), then it is mapped by  $g$  to the word

$$g(w) = [X_0, a_1][X_{i_2}, a_2] \cdots [X_{i_n}, a_n, X_0].$$

We now define the substitution  $\Sigma$  according to the following rules. The derivation starts with a rule of the set

$$\{S \rightarrow g(w) \mid w \in A\}.$$

Then the derivation continues by simulating an insertion. This is done by the rules of the set

$$\begin{aligned} & \{[X_i, a] \rightarrow [X_i, a] \mid [X_i, a] \in N_1, I_i = \emptyset\} \\ & \cup \{[X_i, a] \rightarrow f(w)[X_i, a] \mid [X_i, a] \in N_1, w \in I_i\} \\ & \cup \{[X_i, a, X_0] \rightarrow [X_i, a, X_0] \mid [X_i, a, X_0] \in N_2, I_i = I_0 = \emptyset\} \\ & \cup \{[X_i, a, X_0] \rightarrow f(w)[X_i, a, X_0] \mid [X_i, a, X_0] \in N_2, w \in I_i, I_0 = \emptyset\} \\ & \cup \{[X_i, a, X_0] \rightarrow [X_i, a]g(w) \mid [X_i, a, X_0] \in N_2, w \in I_0, I_i = \emptyset\} \\ & \cup \{[X_i, a, X_0] \rightarrow f(u)[X_i, a]g(v) \mid [X_i, a, X_0] \in N_2, u \in I_i, v \in I_0\}. \end{aligned}$$

The termination is done by a rule of the set

$$\{[X_i, a] \rightarrow a \mid [X_i, a] \in N_1, a \in \Sigma\} \cup \{[X_i, a, X_0] \rightarrow a \mid [X_i, a, X_0] \in N_2, a \in \Sigma\}.$$

In order to prevent a word from losing insertion positions by terminating at some position but not terminating at another position, the terminal symbols are changed to failure symbols which leads to generating a word which does not belong to the language generated. This is achieved by the rules of the set

$$\{a \rightarrow F \mid a \in \Sigma\} \cup \{F \rightarrow F\}.$$

This completes the description of the E0L-system  $\Gamma$ .

The idea behind the symbols acting as nonterminals is that any bond word generated by the system  $\gamma$  can be encoded by a string of nonterminals. For instance, the bond word  $[{}_0 a \text{ } _{-1}] [{}_1 b \text{ } _{-2}] [{}_2 c \text{ } _0]$  will be encoded as  $[X_0, a][X_1, b][X_2, c, X_0]$ . Then by construction, it is not hard to see that any parallel insertion of words from  $I$  into the bond word can be simulated by an application of appropriate rules described above at the correct corresponding position within the sentential form derived by the E0L-system. The bond erasing homomorphism is simulated by simultaneously applying rules of the form  $[X_i, a] \rightarrow a$  and  $[X_i, a, X_0] \rightarrow a$ . If such a rule is applied for some but not every nonterminal, then a sentential form is produced which does not correspond to a word generated by the bPINS-system  $\gamma$ . But neither this sentential form is a word of the language generated by the E0L-system  $\Gamma$  nor any derivative of it because in the next step, the failure symbol  $F$  will occur which cannot disappear again. The details of the proof showing that  $L(\Gamma) = L(\gamma)$  are left to the reader.

We have now shown that  $\mathcal{L}(\text{bPINS}) \subseteq \mathcal{L}(\text{E0L})$ . The properness of the inclusion follows from Lemma 4.14 with the witness language  $L = \{a^{2^n} b^{n-1} c \mid n \geq 1\}$ .

We now prove the incomparability results. From Lemma 4.14, we already know that there is a language which is generated by an L-system of any kind but which is not generated by a bPINS-system. Due to the inclusion relation, we now only have to show that there is a language

$$L_1 \in \mathcal{L}(\text{bPINS}) \setminus \mathcal{L}(\text{T0L})$$

and a language

$$L_2 \in \mathcal{L}(\text{bPINS}) \setminus \mathcal{L}(\text{EDT0L}).$$

Let  $L_1 = \{a, aa\}$ . This language cannot be generated by a T0L-system (see [23]) but it is finite and can therefore be generated by a bPINS-system.

Let  $L_2 = \{w \mid w \in \{a, b\}^*, \exists n \geq 0 : |w| = 2^n\}$ . This language cannot be generated by an EDT0L-system (see [23]) but it can be generated by a bPINS-system as the following construction shows.

Let  $\gamma = (\{a, b\}, A, I)$  be a bPINS-system with

$$A = \{[{}_0 x \text{ } _0] \mid x \in \{a, b\}\} \cup \{[{}_0 x \text{ } _{-1}] [{}_1 y \text{ } _0] \mid x \in \{a, b\}, y \in \{a, b\}\}$$

and

$$I = \{[{}_1 x \text{ } _{-2}] [{}_2 y \text{ } _{-1}] \mid x \in \{a, b\}, y \in \{a, b\}\}.$$

The words of the set  $A$  yield all the words of the language  $L_2$  with length  $2^0$  or  $2^1$ . The words of length  $2^1$  have one position where an insertion can be applied. By inserting, all the words of the language  $L_2$  with length  $2^2$  are obtained and each of these words has two positions where an insertion can be applied. In each derivation step, at every possible insertion position, two letters are inserted. Hence, from the words of length  $2^n$  for some natural number  $n \geq 1$  (which always have  $2^{n-1}$  possible insertion positions), all the words of the language  $L_2$

with length  $2^{n+1}$  are obtained and each of these words has  $2^n$  positions where an insertion can be applied. Other words are not generated. Thus, the bPINS-system  $\gamma$  generates the language  $L_2$ .

Hence, the languages  $L$ ,  $L_1$  and  $L_2$  are the witness languages for the incomparability results.  $\square$

For any natural number  $n \geq 1$ , let  $\mathcal{L}(\text{bPINS}_n)$  denote the family of all languages which are generated by a bPINS-system where at most  $n$  different bonds occur in the words (axioms and insertion strings). These families form an infinite hierarchy.

**Theorem 4.16.** *For any natural number  $n \geq 1$ , we have the strict inclusion*

$$\mathcal{L}(\text{bPINS}_n) \subset \mathcal{L}(\text{bPINS}_{n+1}).$$

*Proof.* The theorem can be proven analogously to the sequential case—see Theorem 4.9—where the bPINS-systems generating the witness languages are obtained according to the construction given in the proof of the theorem stating that any sequential system can be simulated by a parallel one (Thm. 4.12).  $\square$

In the proof of Theorem 4.12, we have constructed a parallel system which simulates a sequential system. In this construction, the number of the bonds does not change. Hence, we obtain the inclusion  $\mathcal{L}(\text{bSINS}_n) \subseteq \mathcal{L}(\text{bPINS}_n)$  for any natural number  $n \geq 1$  with the same proof. Moreover, this inclusion is strict.

**Theorem 4.17.** *For any natural number  $n \geq 1$ , we have the strict inclusion*

$$\mathcal{L}(\text{bSINS}_n) \subset \mathcal{L}(\text{bPINS}_n).$$

*Proof.* The inclusion can be shown analogously to Theorem 4.12.

As a witness language for the properness, we take  $\{a^{2^n-1} \mid n \geq 1\}$ . It can be generated by the bPINS-system  $\gamma = (\{a\}, A, I)$  with  $A = I = \{[_0 a _0]\}$  which uses only one bond. According to Theorem 4.16, this language belongs to any family  $\mathcal{L}(\text{bPINS}_n)$  for  $n \geq 1$ . Since the language is unary but not regular, it does not belong to any family  $\mathcal{L}(\text{bSINS}_n)$  for  $n \geq 1$  by Corollary 4.8.  $\square$

Since the witness languages used in the proof of Theorem 4.16 are all regular and belong to the family  $\mathcal{L}(\text{bSINS})$  (by Cor. 4.3), we immediately see that the family  $\mathcal{L}(\text{bSINS})$  is not included in any of the families  $\mathcal{L}(\text{bPINS}_n)$ . On the other hand, the language  $\{a^{2^n-1} \mid n \geq 1\}$  belongs to the family  $\mathcal{L}(\text{bPINS}_1)$  (proof of Thm. 4.17) and, by Theorem 4.16, also to any family  $\mathcal{L}(\text{bPINS}_n)$  with  $n \geq 1$ . According to Corollary 4.7, however, this language does not belong to the family  $\mathcal{L}(\text{bSINS})$ .

**Corollary 4.18.** *The families  $\mathcal{L}(\text{REG})$  and  $\mathcal{L}(\text{bSINS})$  are incomparable to any family  $\mathcal{L}(\text{bPINS}_n)$  for  $n \geq 1$ .*

Similarly, we also obtain the following result.

**Corollary 4.19.** *For each natural number  $n \geq 1$ , the two families  $\mathcal{L}(\text{bPINS}_n)$  and  $\mathcal{L}(\text{bSINS}_{n+1})$  are incomparable.*

*Proof.* The witness language for the strict inclusion  $\mathcal{L}(\text{bPINS}_n) \subset \mathcal{L}(\text{bPINS}_{n+1})$  (Thm. 4.16) is also a witness for the strict inclusion  $\mathcal{L}(\text{bSINS}_n) \subset \mathcal{L}(\text{bSINS}_{n+1})$  (Thm. 4.9). Hence, this language belongs to the set  $\mathcal{L}(\text{bSINS}_{n+1}) \setminus \mathcal{L}(\text{bPINS}_n)$ . On the other hand, the language  $\{a^{2^n-1} \mid n \geq 1\}$  belongs to the family  $\mathcal{L}(\text{bPINS}_1)$  (proof of Thm. 4.17) and, by Theorem 4.16, also to any family  $\mathcal{L}(\text{bPINS}_n)$  with  $n \geq 1$ . According to Corollary 4.7, however, this language does not belong to the family  $\mathcal{L}(\text{bSINS})$  and, hence, to no family  $\mathcal{L}(\text{bSINS}_{n+1})$ .  $\square$

We finally prove the incomparability of the families  $\mathcal{L}(\text{CF})$  and  $\mathcal{L}(\text{bPINS})$  which was left open in the conference version [4].

**Theorem 4.20.** *The families  $\mathcal{L}(\text{CF})$  and  $\mathcal{L}(\text{bPINS})$  are incomparable.*

*Proof.* In Example 3.4, we have seen that the language  $\{a^{2^n} \mid n \geq 0\}$  is generated by a bPINS-system. This language is not context-free. Hence,  $\mathcal{L}(\text{bPINS}) \not\subseteq \mathcal{L}(\text{CF})$ . As a witness language for the relation  $\mathcal{L}(\text{CF}) \not\subseteq \mathcal{L}(\text{bPINS})$ , we take the language  $L = \{a^n b^n \mid n \geq 0\}$ . This language is context-free. We now prove that this language cannot be generated by a bPINS-system.

Assume that the language  $L$  can be generated by a bPINS-system

$$\gamma = (\{a, b\}, A, I).$$

Let  $k$  be the number of axioms and let  $A_1, A_2, \dots, A_k$  be the axioms. Further, we eliminate all insertion strings from the set  $I$  which can never be used (which have outer bonds not occurring in any word derived from an axiom).

We first observe that no insertion string contains the letters  $a$  and  $b$  at the same time. Otherwise it would contain the subword  $ab$  or  $ba$ . An insertion in the latter case would immediately lead to a word which does not belong to the language  $L$ . An insertion in the former case would produce at least two occurrences of the same bonds where this insertion can take place. A further insertion of the same word would generate a word which contains two occurrences of the subword  $ab$  and this word also does not belong to the language  $L$ . Hence, every insertion word consists either of the letter  $a$  or of the letter  $b$  only.

Moreover, let  $[{}_i x^n -_i] \in I$  be an insertion string for some letter  $x \in \{a, b\}$ . Then, for every insertion string  $[{}_i y^m -_i] \in I$  with  $y \in \{a, b\}$  and the same outer bonds, we know that  $x = y$  because otherwise it would be possible to generate a word which does not have the required form and also  $m = n$  because the insertion of the word  $[{}_i x^n -_i]$  would lead to a different word than the insertion of the word  $[{}_i y^m -_i]$  if all other insertions are the same in both cases and, moreover, in at least one of these cases, a word would be generated which does not have the same number of the letter  $a$  and the letter  $b$  which is also a contradiction.

For the same reason, we even observe that, from each axiom, the derivation of words is ‘length deterministic’ (all words generated in some derivation step have the same length, there can be different words with respect to inner bonds, but all the words derived from such words again have the same length). Therefore, for each axiom  $A_i$  with  $1 \leq i \leq k$ , there is a unique sequence  $l_{i,0}, l_{i,1}, l_{i,2}, \dots$  such that, for every derivation  $A_i \Rightarrow_{\gamma}^j w$  in  $j$  steps ( $j \geq 0$ ), the length of the derived word  $w$  is equal to  $l_{i,j}$ . For each axiom  $A_i$  ( $1 \leq i \leq k$ ), we have one of two possibilities: Either no insertion string can be inserted, then  $l_{i,j} = |A_i|$  for every number  $j$  with  $j \geq 0$ , or there is an insertion string applicable, then the number of bonds where it can be inserted is increased in each derivation step by at least the factor two. Thus, in this case, the difference between any two lengths  $l_{i,j}$  and  $l_{i,j+1}$  is increasing exponentially:  $l_{i,j+1} \geq 2^j + l_{i,j}$  for  $j \geq 0$ . Consider

$$m = \max\{l_{i,k} \mid 1 \leq i \leq k\}.$$

Can each word with a length  $l$  where

$$m < l < m + 2^k$$

be generated by the system?

For each axiom  $A_i$  with  $1 \leq i \leq k$  which produces an infinite set, we consider now that number  $r$  for which  $l_{i,r} > m$  and  $l_{i,j} \leq m$  hold for all steps  $j$  with  $j < r$ . Since  $r \geq k$  and  $l_{i,r+1} \geq 2^r + l_{i,r}$ , we obtain

$$l_{i,r+1} \geq 2^k + l_{i,r} > 2^k + m.$$

Hence, each axiom produces at most one word whose length  $l$  is between  $m$  and  $m + 2^k$ . Furthermore, any axiom  $A_i$  where  $l_{i,k} = m$  holds does not yield a word with such a length. Thus, from the  $k$  axioms, at most  $k - 1$  words can be generated which have a length between  $m$  and  $m + 2^k$ . Since  $k - 1 < 2^k - 1$  for all  $k \geq 1$ , the answer to the above question is “no.” Hence, the system does not generate every word of the language. From this contradiction follows that the language  $L$  cannot be generated by a bPINS-system.  $\square$

## 5. CLOSURE PROPERTIES

In this section, we briefly discuss some closure properties of the families of languages generated by bonded sequential and parallel insertion systems. We start our investigation with the family of languages generated by bonded sequential insertion systems.

**Theorem 5.1.** *The family  $\mathcal{L}(\text{bSINS})$  is closed under union and concatenation.*

*Proof.* Let  $\gamma_i = (\Sigma_i, A_i, I_i)$ , for  $i = 1, 2$ , be a bonded sequential insertion system, which is in normalform according to Theorem 4.5. Recall that because of the normalform *no* neutral insertion words, that are words with the external bond 0, exist. Moreover, we may assume without loss of generality that the bonds used in both systems are distinct, except for the bond value 0, which only appears in the axiom words on the left and right, respectively. Then we argue as follows:

**Union:** Define  $\gamma = (\Sigma, A, I)$  with  $\Sigma = \Sigma_1 \cup \Sigma_2$ , axiom set  $A = A_1 \cup A_2$  and set of insertion strings  $I = I_1 \cup I_2$ . Then it is easy to see that  $L(\gamma) = L(\gamma_1) \cup L(\gamma_2)$ , since by construction, we either simulate a derivation of the bonded sequential system  $\gamma_1$  or  $\gamma_2$  depending whether we start with an axiom from  $A_1$  or  $A_2$ . Thus, the family  $\mathcal{L}(\text{bSINS})$  is closed under union.

**Concatenation:** Set  $\gamma = (\Sigma, A, I)$  with  $\Sigma = \Sigma_1 \cup \Sigma_2$ . The set of axioms is defined to be  $A = A_1 \cdot A_2$  and the set of insertion words is equal to  $I = I_1 \cup I_2$ . Since we are working with a bonded sequential system, the derivation can be restructured such that first one only uses insertion words from  $I_1$  and finally of  $I_2$ . Thus, it is not hard to see that the constructed bonded sequential insertion system  $\gamma$  satisfies  $L(\gamma) = L(\gamma_1) \cdot L(\gamma_2)$ . The tedious details are left to the reader. Therefore, the family  $\mathcal{L}(\text{bSINS})$  is closed under concatenation, too.

This proves the stated claim. □

Next we consider the operation of intersection with regular sets. There we find the following non-closure result.

**Theorem 5.2.** *The family  $\mathcal{L}(\text{bSINS})$  is not closed under intersection with regular sets.*

*Proof.* In the proof of Corollary 4.3, a bonded sequential insertion system  $\gamma$  is constructed that generates the Dyck language on one parenthesis pair  $a$  and  $b$  (including the empty word). But then  $L(\gamma) \cap a^*b^* = \{a^n b^n \mid n \geq 0\}$ , which is not generated by any bonded sequential insertion system. This is seen as follows: assume to the contrary that the language  $L = \{a^n b^n \mid n \geq 0\}$  is accepted by a bonded sequential insertion system. Then we consider the following cases for the insertion words—we assume that all insertion words are productive, which means that there is a derivation that uses the insertion word under consideration; otherwise unproductive words can be deleted from the insertion set:

1. The insertion word  $v$  is build by a single letter only. Assume that this letter is  $a$ . But then applying the insertion word  $v$  twice in sequence in a derivation, leads to a word where the number of  $a$ 's is not equal to the number of  $b$ 's anymore. The double application of  $v$  in a derivation is possible, because the outer bonds fit. A similar argumentation applies if the word  $v$  is build by the single letter  $b$  only.
2. The insertion word  $v$  contains both the letters  $a$  and  $b$ . But then, we have  $h_{\text{be}}(v) \in a^+b^+$ , because otherwise the generated word is not of the appropriate form of a block of  $a$ 's followed by a block of  $b$ 's. But anyhow applying the insertion word  $v$  twice in sequence in a derivation using the outer bounds of the insertion word leads to a word that is not of the appropriate form  $a^*b^*$  anymore as above.

Since no other case applies, this shows that  $L \notin \mathcal{L}(\text{bSINS})$ . Therefore, the family  $\mathcal{L}(\text{bSINS})$  is not closed under intersection with regular sets. □

As a byproduct from the previous proof, we obtain our next result, because the regular language  $a^*b^*$  belongs also to the language family  $\mathcal{L}(\text{bSINS})$ .

**Corollary 5.3.** *The family  $\mathcal{L}(\text{bSINS})$  is not closed under intersection.*

In the remainder of this section, we investigate the closure properties of the language family  $\mathcal{L}(\text{bPINS})$ . We have to leave open, whether  $\mathcal{L}(\text{bPINS})$  is closed under union, but we conjecture that it is not, but we can show that this language family is not closed under concatenation.

**Theorem 5.4.** *The family  $\mathcal{L}(\text{bPINS})$  is not closed under concatenation.*

*Proof.* Consider the languages

$$L_1 = \{a^{2^n} \mid n \geq 0\} \quad \text{and} \quad L_2 = \{b^{2^m} \mid m \geq 0\},$$

which both belong to  $\mathcal{L}(\text{bPINS})$  as shown in Example 3.4. We show that the language  $L = L_1 \cdot L_2$  cannot be generated by any bonded parallel insertion system. Assume to the contrary that there is such an insertion system  $\gamma = (\{a, b\}, A, I)$  with  $L = L(\gamma)$ . It is easy to see that any bonded parallel insertion system that generates  $L$  must have the property that no insertion word contains both the letters  $a$  and  $b$ . Otherwise we obtain a contradiction since one can derive words that are not of the form  $a^*b^*$  anymore. This property also applies to the bonded parallel insertion system  $\gamma$ . Thus, every insertion word consists either of  $a$ 's or  $b$ 's only. Next we define some important parameters of  $\gamma$ : set

$$n = \max\{|u|_a, |u|_b \mid u \in A\}$$

and

$$\ell = \max\{|v|_a, |v|_b \mid v \in I\}.$$

Then define  $m := \max\{3, n, \ell\}$ . Next we consider the word  $w = a^{2^{m^3 m^3 + 1}} b^{2^{m^3}}$  which is a member of  $L(\gamma)$ . By the choice of the parameters of  $w$  we know that

1. in order to derive the numbers of  $a$ 's in  $w$  we need at least  $m^3 + 1$  non-trivial derivation steps w.r.t. the letters  $a$  since after  $m^3 + 1$  non-trivial derivation steps the number of  $a$ 's is at most  $m^{3m^3 + 1} \leq 2^{m^3 m^3 + 1}$  and
2. in order to derive the number of  $b$ 's in  $w$  at least one non-trivial derivation step w.r.t. the letter  $b$  is needed, since after one non-trivial derivation step the number of  $b$ 's is at most  $m + (m + 1) \cdot m \leq 3m^2 \leq m^3 \leq 2^{m^3}$ .

Thus, in the overall derivation sequence, we find a situation, where the numbers of  $b$ 's in the sentential form is not strictly increasing anymore, because in a bonded system one cannot delete letters by insertion. Using all potential  $b$ -word suffixes of the right form leads to exactly  $m^3$  possible suffixes—the suffixes are  $b, b^2, b^4, \dots, b^{m^3}$ . Because at least one non-trivial derivation w.r.t. the letter  $b$  is performed, there must be a bond  $i$  that is present from the very beginning, where an insertion of a non-empty word  $v$  originally took place. On the other hand, since the number of  $b$ 's cannot strictly increase anymore, there must be trivial bond words in  $I$  for all bonds present in the sentential form. In particular, this also holds for the bond  $i$ . But then we can continue the derivation under consideration by doing a further derivation step now using the trivial bond words for all bonds, except for bond  $i$ , where we use the word  $v$  of length at most  $m$ . Then the number of  $b$ 's derived is  $2^{m^3} + m$ , but this is not a power of two because

$$2^{m^3} < 2^{m^3} + m < 2^{m^3} + 2^{m-1} \leq 2^{m^3} + 2^{m^3} = 2^{m^3+1},$$

since  $m < 2^{m-1}$ , for  $m \geq 3$ . This is a contradiction to our assumption that the bonded parallel insertion system  $\gamma$  generates  $L$ . Hence the language  $L$  cannot be generated by any bonded parallel insertion system. Thus,  $L \notin \mathcal{L}(\text{bPINS})$ .  $\square$

TABLE 1. Closure properties of the languages families  $\mathcal{L}(\text{bSINS})$  and  $\mathcal{L}(\text{bPINS})$ .

Operation	Language family	
	$\mathcal{L}(\text{bSINS})$	$\mathcal{L}(\text{bPINS})$
Union	Yes	?
Concatenation	Yes	No
Intersection	No	No
Intersection with regular sets	No	No

Finally, we consider the family  $\mathcal{L}(\text{bPINS})$  and the intersection with regular sets. Here we also find a non-closure result.

**Theorem 5.5.** *The family  $\mathcal{L}(\text{bPINS})$  is not closed under intersection with regular sets.*

*Proof.* Here we argue as in the proof of Theorem 5.2. A slight modification of the bonded sequential insertion system  $\gamma$  results in a bonded parallel insertion system  $\gamma'$  that generates the Dyck language on one parenthesis pair  $a$  and  $b$ , too. The modification consists by adding the insertion word  $[_0 \lambda _0]$ , to  $\gamma$ —this allows one to postpone insertions at certain places to overcome the parallel restriction in order to mimic a sequential derivation. Thus, we obtain the language

$$L(\gamma') \cap a^*b^* = \{a^n b^n \mid n \geq 0\},$$

which is a language that cannot be generated by any bonded parallel insertion system as shown in the proof of Theorem 4.20. This proves the stated claim on the non-closure of the family  $\mathcal{L}(\text{bPINS})$  under intersection with regular languages.  $\square$

By the previous proof we obtain the next corollary.

**Corollary 5.6.** *The family  $\mathcal{L}(\text{bPINS})$  is not closed under intersection.*

We summarize our results on the closure properties of the language families  $\mathcal{L}(\text{bSINS})$  and  $\mathcal{L}(\text{bPINS})$  in Table 1.

## 6. CONCLUSIONS

In this paper, we have introduced the concept of bonded insertion systems by attaching bonds onto symbols in the alphabet. A bonding alphabet is formed, wherein the systems of bonded sequential insertion and bonded parallel insertion are constructed. From there, we have obtained some results on the generative power of the aforementioned new systems. Firstly, the generative power of bonded sequential insertion systems has been shown to be higher compared to regular grammars but lower compared to context-free grammars, where the relation

$$\mathcal{L}(\text{REG}) \subset \mathcal{L}(\text{bSINS}) \subset \mathcal{L}(\text{CF})$$

was proven. Furthermore, it was also shown that bonded sequential insertion systems and controlled sequential insertion systems are almost the same. On the other hand, we have shown that bonded parallel insertion systems have a higher generative power than bonded sequential insertion systems, where the strict inclusion  $\mathcal{L}(\text{bSINS}) \subset \mathcal{L}(\text{bPINS})$  was shown. Furthermore, we have proven that  $\mathcal{L}(\text{bPINS})$  is strictly included in the family  $\mathcal{L}(\text{EOL})$  of the languages generated by extended Lindenmayer systems without interaction and that it is incomparable to the language families of L-systems of other restrictions. A relation of  $\mathcal{L}(\text{bPINS})$  to the hierarchy of the language families defined by L-systems has been constructed and proven to hold true. As a summary,

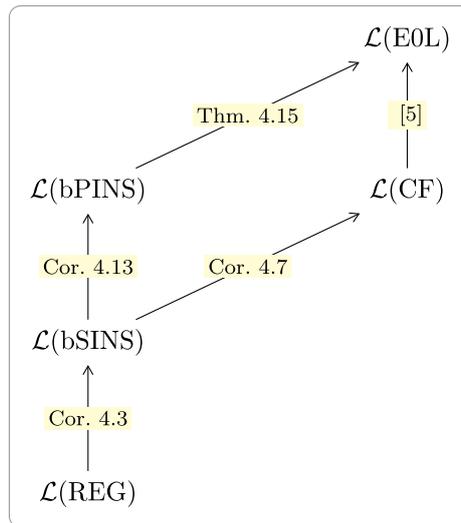


FIGURE 4. Hierarchy of language families.

the hierarchy of language families relating  $\mathcal{L}(\text{REG})$ ,  $\mathcal{L}(\text{bSINS})$ ,  $\mathcal{L}(\text{bPINS})$ ,  $\mathcal{L}(\text{CF})$  and  $\mathcal{L}(\text{EOL})$  is illustrated in Figure 4. An arrow from an entry  $X$  to an entry  $Y$  represents the proper inclusion  $X \subset Y$ .

We additionally investigated the hierarchy which is obtained when the number of different bonds on the symbols in a system is limited.

It still remains to characterize more precisely the language families under consideration. Furthermore, one can consider bonded insertion systems which do not work purely sequentially but also not maximally parallel. In [25], we introduced so-called bonded Indian-parallel insertion systems and bonded uniformly parallel insertion systems. Research on the descriptonal complexity of the family of languages generated by bonded insertion systems is another idea. Further, it remains to investigate bonded deletion systems.

*Acknowledgements.* The last author was supported by MOHE (the Ministry of Higher Education Malaysia) with a MyBrainSC scholarship. We thank the anonymous reviewers for their suggestions.

## REFERENCES

- [1] B. Cui, L. Kari and S. Seki, Block insertion and deletion on trajectories. *Theoret. Comput. Sci.* **412** (2011) 714–728.
- [2] M. Daley, L. Kari, G. Gloor and R. Siromoney, Circular contextual insertions/deletions with applications to biomolecular computation, in *SPIRE/CRIWG, Cancun, Mexico, September 21–24, 1999*. IEEE Computer Society Press (1999) 47–54.
- [3] D. Haussler, Insertion languages. *Inform. Sci.* **31** (1983) 77–89.
- [4] W. Heng Fong, M. Holzer, B. Truthe, S. Turaev and A. Firdaus Yosman, On bonded sequential and parallel insertion systems, in *Proc. 8th Workshop on Non-Classical Models of Automata and Applications (NCMA), Debrecen, Hungary, August 29–30, 2016*, edited by H. Bordihn, R. Freund, B. Nagy and G. Vaszil. Vol. 321 of *books@ocg.at*. Österreichische Computer Gesellschaft, Austria (2016) 163–178.
- [5] G.T. Hermann and G. Rozenberg. *Developmental Systems and Languages*. North-Holland/American Elsevier, 1975.
- [6] M. Ito, L. Kari and G. Thierrin, Insertion and deletion closure of languages. *Theoret. Comput. Sci.* **183** (1997) 3–19.
- [7] M. Ito, L. Kari and G. Thierrin, Shuffle and scattered deletion closure of languages. *Theoret. Comput. Sci.* **245** (2000) 115–133.
- [8] L. Kari, *On insertion and deletion in formal languages*. Ph. D. thesis, University of Turku (1991).
- [9] L. Kari, Insertion and deletion of words: determinism and reversibility, in *Mathematical Foundations of Computer Science*. Vol. 629 of *Lect. Notes Comput. Sci.* Springer (1992) 315–327.
- [10] L. Kari, Generalized derivatives. *Fund. Inform.* **18** (1993) 61–77.
- [11] L. Kari, Insertion operation: closure properties. *Eur. Assoc. Theoret. Comput. Sci. Bull.* **51** (1993) 181–191.
- [12] L. Kari, Deletion operations: closure properties. *Int. J. Comput. Math.* **52** (1994) 23–42.
- [13] L. Kari, Power of controlled insertion and deletion, in *Proc. of Results and Trends in Theoretical Computer Science, Colloquium in Honor of Arto Salomaa, Graz, Austria, June 10–11, 1994*. Vol. 812 of *Lect. Notes Comput. Sci.* Springer (1994) 197–212.

- [14] L. Kari, A. Mateescu, G. Păun and A. Salomaa, Deletion sets. *Fund. Inform.* **19** (1993) 355–370.
- [15] L. Kari, A. Mateescu, G. Păun and A. Salomaa, On parallel deletions applied to a word. *RAIRO: ITA* **29** (1995) 129–144.
- [16] L. Kari, G. Păun, G. Thierrin and S. Yu, At the crossroads of dna computing and formal languages: characterizing re using insertion–deletion systems, in *Proc. of 3rd DIMACS Workshop on DNA Based Computing* (1997) 318–333.
- [17] L. Kari and S. Seki, Schema for parallel insertion and deletion: revisited. *Int. J. Found. Comput. Sci.* **22** (2011) 1655–1668.
- [18] L. Kari and P. Sosík, Aspects of shuffle and deletion on trajectories. *Theoret. Comput. Sci.* **332** (2005) 47–61.
- [19] L. Kari and G. Thierrin, Contextual insertions/deletions and computability. *Inform. Comput.* **131** (1996) 47–61.
- [20] A. Krassovitskiy, *Complexity and modeling power of insertion-deletion systems*. Ph. D. thesis, Universitat Rovira i Virgili (2011).
- [21] G. Păun, G. Rozenberg and A. Salomaa, DNA Computing: New Computing Paradigms. Springer-Verlag (1998).
- [22] J.B. Reece, L.A. Urry, M.L. Cain, S.A. Wasserman, P.V. Minorsky and R.B. Jackson, *Campbell Biology*. 10th ed. Pearson Education Inc. (2011).
- [23] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L-systems*. Academic Press (1980).
- [24] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages*. Springer-Verlag, Berlin (1997).
- [25] A. Firdaus Yosman, M. Holzer, B. Truthe, W. Heng Fong and S. Turaev, On bonded Indian and uniformly parallel insertion systems and their generative power. *Mal. J. Fund. Appl. Sci.* **13** (2017) 769–773.