Language-Theoretic Aspects of String Replication

Valeria Mihalache
Arto Salomaa

Turku Centre for Computer Science
TUCS Technical Report No 51
October 1996
ISBN 951-650-851-0
ISSN 1239-1891

Abstract

The so-called *replication* operation is studied in the case when several insertion contexts consisting of letters are used, as well as in the case when this operation is applied to several strings. Some striking differences, depending on the technical details of the setup, are observed in the generative power of this operation.

TUCS Research Group
Mathematical Structures of Computer Science

1 Introduction

The *replication* operation was introduced in [3] as a new method of generating strings by insertion under some special restrictions. The overall aim was that certain properties of strings are preserved or inherited in the process. One begins with a set of strings, called *replication strings*, and a finite set of pairs of strings, called *insertion contexts*. If w is a replication string and (u, v) is an insertion context such that uv appears as a substring of w, then one can insert in-between u and v any substring of w which starts with v and ends with u. That way a string having one more occurrence of uv as a substring results. The main idea of this operation is depicted in Figure 1.

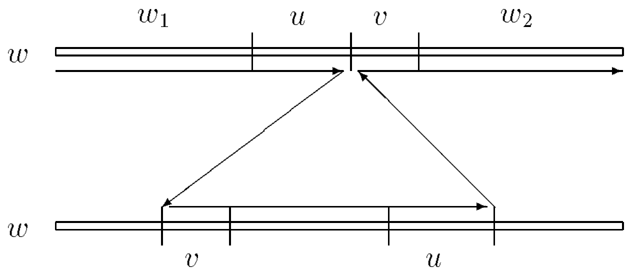

Figure 1

Several variants arise when considering restrictions on the strings to be inserted (that is, maximality, minimality, leftmost, etc.).

From the point of view of formal language theory, this operation is a variant of the customary operation of insertion. It is also related to (Marcus) contextual grammars and to Galiukschov grammars, that way being linguistically motivated. Following the ideas in [1] and [2], we also want to suggest the suitability of this operation for modeling phenomena in molecular biology, such as DNA recombination under the action of specific enzymes, or viral infection within a cell.

DNA and RNA molecules can be regarded as strings (eventually two-leveled) over a four letter alphabet encoding nucleotides, $\{A, T, C, G\}$. It is well known that DNA exists in a circular form, as well as in a linear form. That was originally the reason for extending the notion of *splicing*, a formal model of the recombinant behaviour of DNA sequences under the influence of restriction enzymes and ligases, from linear to circular strings, as well as to the mixed case (combination between a linear and a circular string). For more about splicing, we refer to [2] and [4]. If one considers the splicing rule $(u, v; u, v)$ being applied to two occurrences of a circular string $\hat{\ }uvz = \hat{\ }vzu$, then the circular string $\hat{\ }vzuvzu = \hat{\ }uvzuvz$ results. The same circular string results also when replicating (by extending the replication operator to circular strings in a similar manner with the extension of the splicing operator to circular strings) $\hat{}uvz = \hat{}vzu$ by the insertion context (u, v). Still note that the reversal of a rule $r = (u, v; u, v)$ – which is considered to be simultaneously applied when splicing circular strings by means of rule r – is in this case just r.

One more case when generating strings by replication is just a particular case of DNA recombination is the so-called *mixed splicing*, which occurs when both linear and circular DNA chains are present. Thus, if $r = (u, v; u, v)$ is a splicing rule, $w = \alpha uv\beta$ is a linear string and $\hat{}vzu = \hat{}uvz$ is a circular string, then splicing according to rule r produces the string $\alpha uvzuv\beta$. The same string results if $w = \alpha uv\beta = \gamma vzu\delta$ when replicating it by means of the insertion context (u, v).

We should mention that these cases of recombination of DNA molecules actually occur, as it was pointed out in [1]. In this fashion, the study of replication can be viewed as a "close up" on some particular types of splicing.

The type of generating strings addressed here is motivated also from the point of view of virology. Viruses contain nucleic acid, either DNA or RNA, and protein. The nucleic acid encodes the genetic information unique for each virus. Moreover, there exist the so-called viroids (meaning "virus like"), which are disease-causing organisms that contain only nucleic acid and have no structural protein. The true infectious part of a virus is its nucleic acid. Isolated viruses possess no metabolic systems, so a virus can multiply only within a cell; it is "living" only in the sense that its genetic material directs its own multiplication; outside its host cell a virus is an inert particle. Therefore, when speaking about a virus, it is better to consider it as being in the presence of other nucleotides, and/or DNA (RNA) chains. Let us suppose that uv (corresponding to an insertion context (u, v)) is a DNA (or RNA) chain of a virus (we refer to it as a virus code). Since the infection of a cell occurs when the number of viruses within it exceeds a permitted bound, it is worth investigating situations which lead to the rise of the number of occurrences of the virus code within a DNA chain, and hence within a cell. One can note that the replication operation could be useful in this respect: it multiplies the occurrences of such a code string within the host string.

Then, from biological point of view, questions of the following type are of interest. What happens when there is one such code and various DNA chains – that corresponds to a set of replication strings and one insertion context? How can a cell be modified under the action of several viruses – that corresponds to one replication string and several insertion contexts? What can be said about the number of occurrences and the rate of multiplication, or, still, the appearance of new viruses (if considering that a code uv possibly becomes infectious when the number of its occurrences within a DNA chain exceeds a permitted bound) – that is, growth functions and saturation with some symbols (strings)?

In this paper we investigate problems related to the first two questions mentioned above. More exactly, we study closure properties with regard to the set of replication strings or to the insertion contexts, in case of insertion contexts consisting of letters only. The results will be formulated in a rather general framework, applicable to a number of more specific cases. We also point out the place of replicating systems among other rewriting systems.

2 Basic Definitions

As general formal language notation, we use: V^* = the free monoid generated by the alphabet V, λ = the empty string, $V^+ = V^* - \{\lambda\}$, $|x|$ = the length of $x \in V^*$, $|x|_a$ = the number of occurrences of $a \in V$ in $x \in V^*$, $Pref(x)$ $(Sub(x), Suf(x))$ = the set of prefixes (subwords, suffixes) of $x \in V^*$. The families of regular, and context-sensitive languages are denoted by REG, CS, respectively. As regards language theory in general, we refer to [5].

Definition 1 *A replicating system is a triple*

$$\sigma = (V, W, I),$$

where V is an alphabet, $W \subseteq V^+$, a set of replication strings, and $I \subseteq V \times V$ is a set of insertion contexts.

Definition 2 *With respect to a replicating system as above, for $x, y \in V^*$ we define the direct replication relation by means of an insertion context $(a, b) \in V \times V$ as*

$$\begin{aligned} x \leadsto y \quad \text{iff} \quad & (1) \quad x = x_1abx_2, \ x_1, x_2 \in V^*, \\ & (2) \quad y = x_1azbx_2, \ \text{for } z = bz' = z''a, \\ & (3) \quad z \in Sub(x). \end{aligned}$$

No restriction is imposed about the position of the substring ab in x (condition (1)) or on the way z was selected from $Sub(x)$ (condition (3)). In fact, the following ten possibilities arise rather naturally when considering restrictions on condition (3) (in all cases, $z = bz' = z''a$; we indicate both the term used and an abbreviation):

1. $z = x$ (*total*; t),

2. $z \in Pref(x)$ (*arbitrary prefix; ap*),

3. $z \in Pref(x)$ and z is maximal (if $x = z_1u_1, z_1 = bz_1' = z_1''a$, then $|z| \geq |z_1|$) (*maximal prefix; Mp*),

4. $z \in Pref(x)$ and z is minimal (if $x = z_1u_1, z_1 = bz_1' = z_1''a$, then $|z| \leq |z_1|$) (*minimal prefix; mp*),

5. $z \in Sub(x)$ and z is leftmost (if $x = u_1zu_2, x = u_1'z_1u_2', z_1 = bz_1' = z_1''a$, then $|u_1| \leq |u_1'|$) (*arbitrary leftmost; al*),

6. $z \in Sub(x)$, z is leftmost and maximal ($x = u_1zu_2$ and if $x = u_1'z_1u_2', z_1 = bz_1' = z_1''a$, then $|u_1| \leq |u_1'|$; moreover, if $x = u_1zu_2 = u_1z_1u_2', z_1 = bz_1' = z_1''a$, then $|z| \geq |z_1|$) (*maximal leftmost; Ml*),

7. $z \in Sub(x)$, z is leftmost and minimal ($x = u_1zu_2$ and if $x = u_1'z_1u_2', z_1 = bz_1' = z_1''a$, then $|u_1| \leq |u_1'|$; moreover, if $x = u_1zu_2 = u_1z_1u_2', z_1 = bz_1' = z_1''a$, then $|z| \leq |z_1|$) (*minimal leftmost; ml*),

8. $z \in Sub(x)$ and z is maximal (if $x = u_1zu_2$ and $x = u_1'z_1u_2', z_1 = bz_1' = z_1''a$, $|u_1'| \leq |u_1|, |u_2'| \leq |u_2|$, then $z = z_1$) (*arbitrary maximal; aM*),

9. $z \in Sub(x)$ and z is minimal (if $x = u_1zu_2$ and $x = u_1'z_1u_2', z_1 = bz_1' = z_1''a$, $|u_1'| \geq |u_1|, |u_2'| \geq |u_2|$, then $z = z_1$) (*arbitrary minimal; am*).

The case in Definition 2 corresponds to

10. $z \in Sub(x)$ (*any* subword, *free; af*).

For $g \in \{t, ap, Mp, mp, al, Ml, ml, af, aM, am\}$, we write $x \rightsquigarrow_g y$ if $x \rightsquigarrow y$ and the restrictions required by the g mode of replication are satisfied. $\rightsquigarrow_g^*$ denotes the reflexive and transitive closure of the relation $\rightsquigarrow_g$.

The language generated by the replication system $\sigma = (V, W, I)$ (usually we indicate only W and I) in the mode g is defined as

$L_g(W, I) = \{z \in V^* \mid$ there exists $w \in W$ such that $w \rightsquigarrow_g^* z$ according to insertion contexts in $I\}$.

By convention, $L_g(W, \emptyset) = W$, for any $W \in V^+$.

SF(g) is the notation for the family of languages of the form $L_g(W, I)$ (SF stands for "snake family"). In some specific cases of this notation, restrictions will be imposed to W and I.

3 Generative capacity

In this section we investigate the relations between snake languages of various types and classical language families. Some rather surprising differences are observed betwen the replication modes. For the case of arbitrary minimal, minimal prefix, minimal leftmost and arbitrary free replication mode, systems with only one insertion context and with a regular set of replication strings have the same power as regular grammars. This actually generalizes a result in [3], where it is proved that in the case of one replication string, the system does not exceed the power of regular grammars, for these modes of replication.

When the replication mode considered is arbitrary prefix or arbitrary leftmost, then non-context-free languages can be generated, while in case of a maximal or a total mode of replication, even non-matrix languages can be obtained.

Still for all the replication modes, the power of the system does not exceed the power of context-sensitive grammars.

Theorem 1 *For any $a, b \in V$ and for any regular language $W \subseteq V^+$, the snake language $L_g(W, \{(a,b)\}) \in REG$, for $g \in \{am,\ mp,\ ml,\ af\}$.*

Proof: We consider first the case of *arbitrary minimal* insertion (that is, $g = am$), and then point out the modifications to be made for the other claims in the statement of the theorem.

$W \in REG$ implies that there exists a deterministic finite automaton

$$M = (Q, V, \delta, q_0, F)$$

such that the language recognized by M is $L(M) = W$. We construct a finite automaton

$$M' = (Q', V, \delta', q_0', F')$$

such that $L(M') = L_g(W, \{(a,b)\})$.

Without loss of generality, we can assume that M is a minimal automaton. We list the minimal paths of M from the initial state to a terminal state. Due to the minimality of the paths, no edge in the transition diagram associated to M occurs twice on such a path, but the paths might still have loops. The finiteness of the set of states Q, together with the minimality restriction on the paths, implies that the number of these paths of M is finite. Moreover, any such path is finite, that is, the number of the states occurring in the path is finite. Let

$$d: \quad p_0, p_1, \ldots, p_n$$

be such a minimal path from q_0 to a final state of M, and let us consider the transitions to be marked by symbols $a_i \in V, 1 \leq i \leq n$, respectively, that is, $\delta(p_{i-1}, a_i) = p_i$, for any $i, 1 \leq i \leq n$.

We enumerate next all the sequences of two consecutive edges marked by a followed by b. More exactly, we enumerate all the indices $i_1, i_2, \ldots, i_r, r \geq 0$, $1 \leq i_1 < i_2 < \ldots < i_r < n$ such that $a = a_{i_q}, b = a_{i_q+1}$ (in other words, $\delta(p_{i_q-1}, a) = p_{i_q}, \delta(p_{i_q}, b) = p_{i_q+1}$). All these i_q represent insertion places: after the state p_{i_q} we can insert in d any sequences of states marked by a string $z = bz' = z''a, z', z'' \in V^*$.

In order to do that, we consider all the pairs $(j_1, k_1), \ldots, (j_t, k_t), t \geq 0$, which identify minimal substrings of the string $a_1a_2\ldots a_n$ of the form $z = bz' = z''a, z', z'' \in V^*$. We then have for any $1 \leq l \leq t$,

$$\begin{aligned}
&z_l = a_{j_l}a_{j_{l+1}}\ldots a_{k_l},\\
&\delta(p_{j_l-1}, a_{j_l}) = p_{j_l}, \ldots, \delta(p_{k_l-1}, a_{k_l}) = p_{k_l},\\
&a_{j_l} = b, a_{k_l} = a.
\end{aligned}$$

To any state p_i we associate a state s_i, and, in addition, for any insertion position i_q and for any pair (j_l, k_l) specified above we consider the new states

$$s^{(l,q)}_{j_l-1}, s^{(l,q)}_{j_l}, \ldots, s^{(l,q)}_{k_l}.$$

The automaton M' to accept $L_{am}(W, \{(a,b)\})$ will be constructed such that it contains a "copy" of the path d, modified by including at the insertion positions i_q loops labelled by strings z_l, for any $q, l, 1 \leq q \leq r, 1 \leq l \leq t$. Thus, the transition function δ' of M' is defined as:

$$\begin{aligned}
s_i &= \delta'(s_{i-1}, a_i),\ 1 \leq i \leq n,\\
s^{(l,q)}_{j_l-1} &\in \delta'(s_{i_q}, \lambda),\ 1 \leq l \leq t, 1 \leq q \leq r,\\
s^{(l,q)}_h &\in \delta'(s^{(l,q)}_{h-1}, a_h),\ j_l \leq h \leq k_l, 1 \leq l \leq t, 1 \leq q \leq r,\\
s_{i_q} &\in \delta'(s^{(l,q)}_{k_l}, \lambda),\ 1 \leq l \leq t, 1 \leq q \leq r.
\end{aligned}$$

We associate to each minimal path from q_0 to a final state in M a construction as outlined above for the path d, in such a way that to any state $p \in Q$ a unique state $s \in Q'$ is associated (that is, if a state $p \in Q$ occurs in two different paths of M, we identify the states associated to it in M'). Still the states of the form $s^{(l,q)}_h$ introduced in Q' are different for different paths of M. Because the number of such paths of the automaton M is finite, the construction is effective.

Q' is the set of all the new states in the described construction. The initial state q'_0 of M' is the state associated to q_0, according to the construction (which is unique, by the observation above), and the set of final states of the new automaton, F', is the set of the states associated to the states in F in the described construction.

We prove in the sequel that $L_{am}(W, \{(a,b)\}) = L(M')$.

For the inclusion $L(M') \subseteq L_{am}(W, \{(a,b)\})$, note that any word accepted by M' is obtained from a word accepted by M (and hence from a word in W) by inserting in-between ab substrings of the form $z = bz' = z''a$, $z', z'' \in V^*$.

For the converse inclusion, observe that the insertion places in any word of $L_{am}(W, \{(a,b)\})$ correspond to the insertion places in the word in W it was obtained from, and also that the strings to be inserted in any word of $L_{am}(W, \{(a,b)\})$ are either strings to be inserted in the word it was derived from of W, or concatenations of such strings. A detailed proof goes by induction on the number of steps in replicating a string.

Thus, $L_{am}(W, \{(a,b)\}) = L(M')$ holds.

For the case of inserting in the *minimal prefix* mode, the only change in the proof is that there is at most one string to be inserted (that is, only one pair (j_1, k_1)), for any minimal path of M as considered above.

For the case of inserting in the *minimal leftmost* mode, the proof goes in a similar way, but the pairs $(j_1, k_1), \ldots, (j_t, k_t)$ identifying the strings to be inserted are determined in such a way that those strings $z_l, 1 \leq l \leq t$, are not necesarilly minimal in the am sense, but they have to be leftmost in the first place (they might be of the form $b\alpha b\beta a$, with $\alpha, \beta \in (V - \{a\})^*$), and there is at most one such substring to be inserted for any string in W.

For the case of *arbitrary free* mode, the changes to be made are also for the strings to be inserted, namely they should be of the form $z_l = bz'_l = z''_l a$, $z'_l, z''_l \in V^*$, z'_l, z''_l not necessarily minimal, z'_l might contain b, z''_l might contain a, but still they are subjected to the restriction that z_l does not contain the sequence ab as a substring. □

Corollary 1 $\mathbf{SF}_g(REG, 1) = REG$, *for* $g \in \{am,\ mp,\ ml,\ af\}$, *where we have denoted by* $\mathbf{SF}_g(REG, 1)$ *the snake family generated in the replication mode* g *starting from a regular set of strings and with one insertion context.*

Proof: The inclusion $\mathbf{SF}_g(REG, 1) \subseteq REG$ is by the preceding theorem. For the converse inclusion, consider L a regular language over the alphabet V and let c, d be symbols not in V. Then $L_g(L, \{(c,d)\}) = L$, and hence also $REG \subseteq \mathbf{SF}_g(REG, 1)$. □

For the other modes of replication, we recall that it was proved in [3], for the case of one replication string and one insertion context, that

i) For any $g \in \{Mp, Ml, aM, t\}$, the family $\mathbf{SF}(g)$ contains non-matrix languages.

ii) For any $g \in \{ap, al\}$, the family $\mathbf{SF}(g)$ contains non-context-free languages.

But because the replication is a length-increasing operation on words, we have

Theorem 2 $\mathbf{SF}(g) \subseteq CS$, *for any replication mode* $g \in \{am, af, al, ap, ml, mp, aM, Ml, Mp, t\}$.

4 Closure Properties

Closure properties with respect to the set of replication strings and with respect to the set of insertion contexts are presented in this section, for the various modes of replication. In most of the cases, the closure under union or intersection does not hold. It does hold in some more restrictive cases.

First, we deal with closure properties with respect to the set of replication strings.

Theorem 3 *Replication is a closure operator with respect to the set of replication strings, that is, it satisfies the properties:*

i) $W \subseteq L_g(W, I)$,

ii) $W_1 \subseteq W_2$ *implies* $L_g(W_1, I) \subseteq L_g(W_2, I)$,

iii) $L_g(L_g(W, I), I) = L_g(W, I)$,

for any sets of replication strings $W_1, W_2, W \subseteq V^+$, *for any set of insertion contexts* $I \subseteq V \times V$ *and for any replication mode* $g \in \{am, af, ml, al, ap, mp, aM, Ml, Mp, t\}$.

Proof: It follows directly from the definition. □

Theorem 4 *For any sets of replication strings* $W_1, W_2 \subseteq V^+$, *for any set of insertion contexts* $I \subseteq V \times V$, *and for any replication mode* $g \in \{am, af, ml, al, ap, mp, aM, Ml, Mp, t\}$, *the relation* $L_g(W_1 \cup W_2, I) = L_g(W_1, I) \cup L_g(W_2, I)$ *holds.*

Proof: It follows directly from the definition of the replication, because the strings derived from different initial strings are not related in any way. □

However, for intersection the closure does not generally hold. More specifically, we have

Theorem 5 *For any sets of replication strings $W_1, W_2 \subseteq V^+$, for any set of insertion contexts $I \subseteq V \times V$, and for any replication mode $g \in \{am, af, ml, al, ap, mp, aM, Ml, Mp, t\}$, the relation $L_g(W_1 \cap W_2, I) \subseteq L_g(W_1, I) \cap L_g(W_2, I)$ holds.*

Proof: Let $w' \in L_g(W_1 \cap W_2, I)$. Then there exists a string $w \in W_1 \cap W_2$ such that $w \rightsquigarrow_g^* w'$ according to insertion contexts in I. This implies $w \in W_1$, $w \in W_2$ and $w \rightsquigarrow_g^* w'$ according to insertion contexts in I, hence $w' \in L_g(W_1, I)$ and $w' \in L_g(W_2, I)$, and therefore w' belongs to their intersection. $\square$

Theorem 6 *For any $g \in \{am, af, ml, al, ap, mp, aM, Ml, Mp, t\}$, there are $W_1, W_2 \subseteq V^+$, and $I \subseteq V \times V$, such that the relation $L_g(W_1, I) \cap L_g(W_2, I) \subseteq L_g(W_1 \cap W_2, I)$ does not hold.*

Proof: Let $W_1 = \{bab\}, W_2 = \{babab\}, I = \{(a, b)\}$. Then $W_1 \cap W_2 = \emptyset$, and therefore $L_g(W_1 \cap W_2, I) = L_g(\emptyset, I) = \emptyset$. However, $bab \rightsquigarrow_g babab$ for any mode $g \in \{am, af, ml, al, ap, mp, aM, Ml, Mp\}$ of replication, and still from $babab$ the derivation can continue. Therefore, $L_g(W_1, I) \cap L_g(W_2, I) = L_g(W_2, I) \neq \emptyset$ for any g as above.

For the t mode of replication, consider $W_1 = \{baba\}, W_2 = \{babababa\}, I = \{(a, b)\}$. With the same observations as above, we obtain that also in this case the claim in the theorem is true. $\square$

We study in the following closure properties of the replication with respect to the insertion contexts, for union and for intersection.

Obviously, if I_1, I_2 are two sets of insertion contexts such that $I_1 \subseteq I_2$, then $L_g(W, I_1) \subseteq L_g(W, I_2)$, for any insertion mode g and for any set of strings W. However, as it will be proved in the sequel, the replication operator is not closed under union, with respect to the sets of insertion contexts, except for a few particular cases.

Theorem 7 *There are singleton sets $W \subseteq V^+$, $I_1, I_2 \subseteq V \times V$, such that for any $g \in \{am, ml, al, Ml, af, aM\}$, the equality $L_g(W, I_1 \cup I_2) = L_g(W, I_1) \cup L_g(W, I_2)$ does not hold.*

Proof: Consider $W = \{bcacdab\}, I_1 = \{(a, b)\}, I_2 = \{(c, d)\}$, where a, b, c and d are distinct symbols. Denote $w = bcacdab$.

One can easily note that $L_g(W, I_2) = \{bcacdab\}$, for any insertion mode g.

In both the cases of the set of insertion contexts being I_1 or $I = I_1 \cup I_2$, the insertion context which can be used to modify w is (a, b), thus resulting in the string $w_1 = bcacdabcab$, in case of the insertion mode $g \in \{am, ml, af, al\}$ or in the string $w_2 = bcacdabcacdab$, in case of the insertion mode $g \in \{Ml, aM, af, al\}$ (note that in case of $g \in \{af, al\}$ we have two alternatives). We have $|w_1| = 10, |w_2| = 13$. If the replication continues from w_1 by the insertion context (a, b) in one of the modes $g \in \{am, af, al\}$, then the minimal resulting string is $w_3 = bcacdabcabcab$, for which $|w_3| = 13$. In the case when the set of insertion contexts is I, then from w_1 we can also obtain, by using the insertion context (c, d), the string $w_4 = bcacdabcdabcab$, for which $|w_4| = 14$, in any of the replicating modes $g \in \{am, ml, af, al, Ml, aM\}$.

Since any word, let it be w', derived from w_3 (in case $g \in \{am, ml\}$), or from w_2 (in case $g \in \{Ml, aM\}$), or from w_2, w_3 (in case $g \in \{af, al\}$), is obtained by inserting at least two symbols, it follows that $|w'| \geq |w_3|+2 = 15$, or $|w'| \geq |w_2| + 2 = 15$. Hence in any situation, the strings in $L_g(W, I_1)$ have length either 7 $(= |w|)$, or 10 $(= |w_1|)$, or 13 $(= |w_2| = |w_3|)$, or greater than or equal to 15. In no case a string of length 14 $(= |w_4|)$ results, hence $w_4 \in L_g(W, I_1 \cup I_2) - L_g(W, I_1)$, and therefore $w_4 \in L_g(W, I_1 \cup I_2) - (L_g(W, I_1) \cup L_g(W, I_2))$, for any $g \in \{am, ml, al, af, Ml, \ aM\}$. $\square$

Theorem 8 *In general, for* $g \in \{ap, mp, Mp\}$, $W \subseteq V^+$ *a set of inserting strings and for two sets of inserting contexts* $I_1, I_2 \subseteq V \times V$, *the equality* $L_g(W, I_1 \cup I_2) = L_g(W, I_1) \cup L_g(W, I_2)$ *does not hold. More precisely, there are singleton sets* $W \subseteq V^+$, $I_1, I_2 \subseteq V \times V$, *such that the inclusion* $L_g(W, I_1 \cup I_2) \subseteq L_g(W, I_1) \cup L_g(W, I_2)$ *does not hold for any* $g \in \{ap, mp, Mp\}$.

Proof: Consider $W = \{babcb\}, I_1 = \{(a, b)\}, I_2 = \{(c, b)\}$, for a, b, c distinct symbols. Denote $w = babcb$.

One can observe that

$$\begin{aligned} L_g(W, I_1) &= (ba)^+bcb, \text{ for } g \in \{ap, mp\}, \\ L_{Mp}(W, I_1) &= \{(ba)^{2^n}bcb \mid n \geq 0\}, \\ L_g(W, I_2) &= (babc)^+b, \text{ for } g \in \{ap, mp\}, \\ L_{Mp}(W, I_2) &= \{(babc)^{2^n}b \mid n \geq 0\}, \end{aligned}$$

while when the set of insertion contexts is $I_1 \cup I_2$ the following derivations are possible:

$$w = babcb \rightsquigarrow_g babcbabcb \rightsquigarrow_g babcbababcb = w', \text{ for } g \in \{am, mp\},$$

where in the first replication step the insertion context (c, b) has been used, and in the second one the context (a, b), as well as the derivation

$$w = babcb \rightsquigarrow_{Mp} babcbabcb \rightsquigarrow_{Mp} bababcbabcbabcb = w'',$$

where again in the first replication step the insertion context (c, a) has been used, and in the second one the context (a, b).

Thus we have obtained $w' \in L_g(W, I_1 \cup I_2) - (L_g(W, I_1) \cup L_g(W, I_2))$, for $g \in \{ap, mp\}$, and $w'' \in L_{Mp}(W, I_1 \cup I_2) - (L_{Mp}(W, I_1) \cup L_{Mp}(W, I_2))$. $\square$

However, in some particular cases, for the prefix replication mode, we have the equality.

Theorem 9 *If $W \subseteq V^+$ and $I_1, I_2 \subseteq V \times V$ are two sets of insertion contexts such that $\{b \mid (a, b) \in I_1\} \cap \{d \mid (c, d) \in I_2\} = \emptyset$, then for any insertion mode $g \in \{ap, mp, Mp\}$ we have $L_g(W, I_1 \cup I_2) = L_g(W, I_1) \cup L_g(W, I_2)$.*

Proof: The inclusion $L_g(W, I_1) \cup L_g(W, I_2) \subseteq L_g(W, I_1 \cup I_2)$ follows by the above observation.

Conversely, let $w' = s\overline{w} \in L_g(W, I_1 \cup I_2)$, where $s \in V, \overline{w} \in V^*$. Then $w \rightsquigarrow_g^* w'$ for a replication mode $g \in \{ap, mp, Mp\}$, and for a string $w \in W$, according to insertion contexts in $I_1 \cup I_2$. Because the replicating mode is a prefix one (and the first symbol is preserved by the replication), it follows that $w = sx, x \in V^*$, and only insertion contexts of the form $(v, s) \in I_1 \cup I_2$ can be applied in the derivation $w \rightsquigarrow_g^* w'$. By the restriction $\{b \mid (a, b) \in I_1\} \cap \{d \mid (c, d) \in I_2\} = \emptyset$ in the hypothesis of the theorem, all these replication contexts (v, s) have to belong either to I_1 or to I_2. Thus, either $w' \in L_g(W, I_1)$, or $w' \in L_g(W, I_2)$, hence $w' \in L_g(W, I_1) \cup L_g(W, I_2)$, which implies $L_g(W, I_1 \cup I_2) \subseteq L_g(W, I_1) \cup L_g(W, I_2)$. $\square$

Also in the case of the total replication mode the closure under union with respect to the insertion contexts holds.

Theorem 10 *For any set of strings $W \subseteq V^+$ and for any sets of insertion contexts $I_1, I_2 \in V \times V$, the equality $L_t(W, I_1 \cup I_2) = L_t(W, I_1) \cup L_t(W, I_2)$ holds.*

Proof: It follows with similar arguments as in the preceding proof, by the observation that the total mode of replication preserves both the first and the last symbol in a word, and hence to a string $w \in W$, as well as to any word obtained from it in the total mode of replication, at most one insertion context (a, b) can be applied, namely for $w = aw'b, w' \in V^*$, that is, $L_t(w, I_1 \cup I_2) = L_t(w, \{(a, b)\} \cap (I_1 \cup I_2)) = L_t(w, \{(a, b)\} \cap I_1) \cup L_t(w, \{(a, b)\} \cap I_2)$. We then obtain $L_t(W, I_1 \cup I_2) = L_t(W, I_1) \cup L_t(W, I_2)$. $\square$

As for the intersection of the sets of insertion contexts, it is easy to see that for any set of strings W, for any two sets of insertion contexts I_1, I_2, and for any replication mode g, we have

$$L_g(W, I_1 \cap I_2) \subseteq L_g(W, I_1) \cap L_g(W, I_2).$$

However, the converse inclusion does not generally hold, but only in a few particular situations.

Theorem 11 *There exist singleton sets $W \subseteq V^+$, $I_1, I_2 \subseteq V \times V$, such that for every $g \in \{am, ml, al, Ml, af, aM\}$, $L_g(W, I_1 \cap I_2) \neq L_g(W, I_1) \cap L_g(W, I_2)$.*

Proof: Consider $W = \{bab\}, I_1 = \{(a, b)\}, I_2 = \{(b, a)\}$. Denote $w = bab$.

If the replication mode is $g \in \{al, ml, af, am\}$, then we can insert the string ba in w, as well as in any word derived from w (in between ab), when the insertion context is (a, b), and we can insert the string ab in w, as well as in any word derived from w (in between ba), when the insertion context is (b, a). That way, we have

$$L_g(w, (a, b)) = L_g(w, (b, a)) = (ba)^+b, \text{ for } g \in \{al, ml, af, am\}.$$

So $L_g(w, (a, b)) \cap L_g(w, (b, a)) = (ba)^+b$. But $L_g(w, \{(a, b)\} \cap \{(b, a)\}) = L_g(w, \emptyset) = \{w\}$ (by convention), and therefore we have $L_g(W, I_1) \cap L_g(W, I_2) \not\subseteq L_g(W, I_1 \cap I_2)$, for $g \in \{al, ml, af, am\}$.

In case $g \in \{aM, Ml\}$, one can observe $L_g(w, (a, b)) = L_g(w, (b, a)) = \{(ba)^{2^n}b \mid n \geq 0\}$. Therefore, $L_g(W, I_1) \cap L_g(W, I_2) = \{(ba)^{2^n}b \mid n \geq 0\}$, while $L_g(W, I_1 \cap I_2) = L_g(W, \emptyset) = \{bab\}$, and hence again $L_g(W, I_1) \cap L_g(W, I_2) \not\subseteq L_g(W, I_1 \cap I_2)$. □

Theorem 12 *In general, for $g \in \{ap, mp, Mp\}$, $W \subseteq V^+$ a set of inserting strings and for two sets of inserting contexts $I_1, I_2 \subseteq V \times V$, the equality $L_g(W, I_1 \cap I_2) = L_g(W, I_1) \cap L_g(W, I_2)$ does not hold. More precisely, for any $g \in \{ap, mp, Mp\}$, there exist at most two-element sets $W \subseteq V^+$, $I_1, I_2 \subseteq V \times V$, such that $L_g(W, I_1) \cap L_g(W, I_2) \not\subseteq L_g(W, I_1 \cap I_2)$.*

Proof: We consider first the case $g \in \{ap, mp\}$. Denote $w_1 = babccbababcb$, $w_2 = bababccb$. Let $W = \{w_1, w_2\}, I_1 = \{(a, b)\}, I_2 = \{(c, b)\}$, for a, b, c distinct symbols.

One can observe that in both of the modes ap and mp of replication, w_1 yields $w' = bababccbababcb$ by the insertion context in I_1, when inserting in-between the leftmost occurrence of ab the string ba, and w_2 yields the same string w' by the insertion context in I_2, when inserting in-between cb the string $bababc$. Still notice that the replication can continue from w' by both of the insertion contexts in I_1, I_2. Therefore, $W \subset L_g(W, I_1) \cap L_g(W, I_2)$, strict inclusion, while $L_g(W, I_1 \cap I_2) = L_g(W, \emptyset) = W$, and hence $L_g(W, I_1) \cap L_g(W, I_2) \not\subseteq L_g(W, I_1 \cap I_2)$.

As for the maximal prefix replication mode, consider $w_1 = bcbcbabcbc$, $w_2 = bcbabcbc$, $W = \{w_1, w_2\}, I_1 = \{(a, b)\}, I_2 = \{(c, b)\}$, where a, b, c are distinct symbols.

One can observe that $w_1 \rightsquigarrow_{Mp} bcbcbabcbcbabcbc = w'$ by the insertion context in I_1, and also $w_2 \rightsquigarrow_{Mp} w'$, by the insertion context in I_2, when inserting in-between the leftmost occurrence of cb the maximal prefix required by the insertion context. Now the relation in the statement of the theorem follows with the same observations as for the cases $g \in \{mp, ap\}$. $\square$

When additional restrictions are imposed to the sets of insertion contexts, in case of a prefix or total replication mode the closure under intersection holds.

Theorem 13 *If $W \subseteq V^+$ and $I_1, I_2 \subseteq V \times V$ are two sets of insertion contexts such that $\{b \mid (a, b) \in I_1\} \cap \{d \mid (c, d) \in I_2\} = \emptyset$, then for any insertion mode $g \in \{ap, mp, Mp\}$ we have $L_g(W, I_1 \cap I_2) = L_g(W, I_1) \cap L_g(W, I_2)$.*

Proof: We only have to prove that $L_g(W, I_1) \cap L_g(W, I_2) \subseteq L_g(W, I_1 \cap I_2)$ for $g \in \{ap, mp, Mp\}$.

Let $w' \in L_g(W, I_1) \cap L_g(W, I_2)$. Then there exist $w_1, w_2 \in W$ such that $w_1 \rightsquigarrow_g^* w'$ by insertion contexts in I_1 and $w_2 \rightsquigarrow_g^* w'$ by insertion contexts in I_2. We consider the first symbol in w', that is $w' = b\overline{w}, b \in V, \overline{w} \in V^*$. Since any of the replication modes preserves the first symbol in the word, it then follows that also w_1 and w_2 have this b as the first symbol. The replication mode considered here is a prefix one, and therefore the derivation $w_1 \rightsquigarrow_g^* w'$ above has to be performed using insertion contexts of the form $(a, b) \in I_1$, and the derivation $w_2 \rightsquigarrow_g^* w'$ above has to be performed using insertion contexts of the form $(c, b) \in I_2$. Due to the restriction in the hypothesis, the only possibility which can occur is that these are zero-step derivations, and hence $w_1 = w' = w_2$.

So $L_g(W, I_1) \cap L_g(W, I_2) = W$, while $L_g(W, I_1 \cap I_2) = L_g(W, \emptyset) = W$, which proves the theorem. $\square$

Theorem 14 *For any set of strings $W \subseteq V^+$ and for any sets of insertion contexts $I_1, I_2 \in V \times V$, the equality $L_t(W, I_1 \cap I_2) = L_t(W, I_1) \cap L_t(W, I_2)$ holds.*

Proof: It can be proved in a similar manner as the preceding theorem, observing once more that the total mode of replication preserves both the first and the last symbol in a word. $\square$

5 Conclusion

We have investigated properties of a new mechanism of generating strings, which can be considered as a study in formal languages and also in combinatorics on words.

The results here should be interesting on their own right in the theory of formal languages. Yet they are also motivated by molecular biology, linguistics, or learning theory, as it was hinted in [3]. Here we have emphasized only their suitability to model phenomena of molecular biology.

Even though the generative capacity, which has been dealt with in section 3, might have no particular meaning for molecular biology, that is not the case anymore for the closure properties. These have a sound interpretation in terms of virus action on the cells.

Thus, the closure under union with respect to the set of replication strings (Theorem 4) states the natural phenomenon that the "action" a virus produces on several DNA chains is the sum of the actions it produces on each DNA chain (the word "action" here should be understood in a more restrictive sense, it refers to the influence that combination of DNA chains has on increasing the number of occurrences of virus codes within a cell). The fact that the closure under intersection with respect to the set of replication strings does not generally hold (Theorems 5 and 6) could be interpreted as the fact that when, in order to produce a disease, a virus has to act upon several DNA chains, then its effect is less than it were if not being subjected to this restriction, but still having the possibility to affect these DNA chains (and then comparing the results – by intersection). Also the fact that closure under union with respect to the insertion contexts does not hold (Theorem 7) has a natural interpretation. It says that if several viruses can act upon a cell at the same time, then their effect is (obviously) more than if they would have acted separately on the cell, one after another.

References

[1] T. Head, Formal Language Theory and DNA: An Analysis of the Generative Capacity of Specific Recombinant Behaviors, *Bull. Math. Biology*, 49(1987), 739-759

[2] T. Head, Gh. Păun, D. Pixton, Languages and DNA Recombination, *The Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.), Springer-Verlag, to appear

[3] V. Mihalache, Gh. Păun, G. Rozenberg, A. Salomaa, Generating Strings by Replication: A Simple Case, submitted

[4] Gh. Păun, Splicing. A Challenge for Formal Language Theorists, *EATCS Bulletin*, 57(1995), 183-195

[5] A. Salomaa, *Formal Languages*, Academic Press, New York, London, 1973

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**