Generating Strings by Replication: A Simple Case

Valeria Mihalache
Gheorghe Păun
Institute of Mathematics of the Romanian Academy
PO Box 1 – 764, 70700 Bucureşti, Romania
Grzegorz Rozenberg
Department of Computer Science, Leiden University
2300 RA Leiden, The Netherlands
and
Department of Computer Science, University of Colorado at Boulder
Boulder, CO 80309, USA
Arto Salomaa

Turku Centre for Computer Science
TUCS Technical Report No 17
June 1996
ISBN 951-650-767-0
ISSN 1239-1891

Abstract

We consider the following new way of generating a language: start with a string w and a given set of *insertion contexts* (u, v); if uv appears as a substring of w, then we can insert in between u and v any substring of w which starts with v and ends with u (hence preserving the link restriction encoded by (u, v)); iterating this operation, we get a language. Many variants can be imagined, concerning the place where the insertion is performed (free, leftmost, etc.), and the string to be inserted (any one, a prefix of the current string, a minimal or a maximal prefix, a minimal or a maximal substring, the whole string, etc.). We investigate here ten such variants, for the case when we have only one insertion context, namely of the form (a, b), where a and b are symbols. The corresponding families of languages are compared to each other (all the relationships are settled), as well as to families in Chomsky hierarchy. A series of topics for further research are pointed out.

TUCS Research Group
Mathematical Structures of Computer Science

1 Introduction

The present paper can be considered as a contribution to both formal language theory and combinatorics on words. We investigate a generative device of a type resembling certain somewhat "exotic" devices in formal language theory, not based on rewriting, as the Chomsky grammars and Lindenmayer systems are, but closer to Marcus grammars [7], [8], or Galiukschov grammars [2]. The main idea is depicted in figure 1.

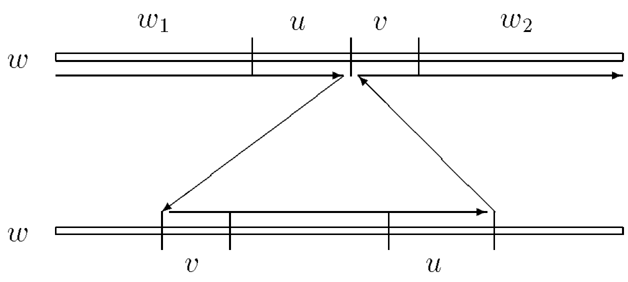

Figure 1

One gives a starting string and a finite set of *insertion contexts* (u, v). They can be considered "weak links", where a string can be broken and relinked after adding a new segment. If the string contains a substring starting with v and ending with u, then such a substring can be inserted between u and v, because the links are preserved, both inside the new segment (because it is a part of the existing string) and between this segment and the neighbouring parts of the string (we have uv, uv at the ends of the added substring). (In figure 1 the string w has been written twice, in order to point out both the insertion place – on the upper position – and the substring to be inserted – on the lower position.) Iterating the procedure, we obtain a language.

We call this operation (controlled) *replication*; sometimes, we say that this is a replication of *snakes*. We investigate here only the case when only one context (u, v) is given, namely with symbols for u and v.

In general, many variants can be considered, depending on the answer to the questions (1) *where* are we allowed to insert ? and (2) *what* can we insert ?

We do not present such variants here, formal definitions can be found in the next section (for the various possibilities of answering question (2)). We want, however, to stress the intimate connection between our "snake replicating machines" and the insertion operation in [6], the insertion grammars of [2] (based on rewriting rules of the form $uv \to uxv$), the splicing and crossover operations specific to DNA recombination [4], [5], as well as to learning theory: we may assume that the strings we work with are representations of certain piece of knowledge, which is self-enriched, longer and longer strings are produced "by internal means", only rearranging and replicating already existing information.

Although our devices look very simple, they raise significant combinatorial questions and we expect that these questions are even difficult for more complex variants of replicating systems (for instance, with insertion contexts of arbitrary length).

2 Simple replicating systems

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 $x \in V^*$, $alph(L) = \{a \mid |x|_a > 0$ for some $x \in L\}$. If $V = \{a_1, \ldots, a_n\}$ and $x \in V^*$, its Parikh vector is $\Psi_V(x) = (|x|_{a_1}, \ldots, |x|_{a_n})$. The mapping Ψ_V is extended in the natural way to languages. The families of finite, regular, context-free and context-sensitive languages are denoted by FIN, REG, CF, CS, respectively. By MAT we denote the family of languages generated by matrix grammars without appearance checking but possibly containing λ-rules. Further elements of formal language theory can be found in [9].

Definition 1. A *simple replicating system* is a triple

$$\sigma = (V, w, (a, b)),$$

where V is an alphabet, $w \in V^+$, and $a, b \in V$.

Definition 2. With respect to a replicating system as above, for $x, y \in V^*$ we define the *direct replication* relation by

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

No restriction is imposed about the place of the substring ab in x (condition (1)) or on the way z was selected from $Sub(x)$ (condition (3)). We discuss here ten posibilities arising from considering restrictions on condition (3).

Namely, condition (3) can be replaced by more restrictive ones, as follows (in all cases, $z = bz' = z''a$):

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

2. $z \in Pref(x)$ (*any 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 $|u_1| \leq |u_1'|$) (*any 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 $z = z_1$) (*any 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 $z = z_1$) (*any minimal; am*).

The case in Definition 2 corresponds to

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

We denote by D the set $\{t, ap, Mp, mp, al, Ml, ml, af, aM, am\}$, of previously defined modes of choosing the inserted string. We stress the fact that in all points 1 – 10, z is a string of the form $z = bz' = z''a$. (When $a = b$, we can have $z = a$, but for $a \neq b$ we have $|z| \geq 2$.)

In Definition 2, the replication is done in the *af* mode. We write this as $x \rightsquigarrow_{af} y$. In general, for a replication choosing the inserted string in the mode $g \in D$, we write $x \rightsquigarrow_g y$.

We denote by $\rightsquigarrow_g^*$ the reflexive and transitive closure of the relation $\rightsquigarrow_g$, $g \in D$. (We call it a *replication chain.*) Then, the *language generated* by the replicating system $\sigma = (V, w, (a, b))$ in the mode $g \in D$ is defined by

$$L_g(\sigma) = \{z \in V^* \mid w \rightsquigarrow_g^* z\}.$$

We denote by $SF(g)$ the family of languages of the form $L_g(\sigma)$, $g \in D$ (SF stands for "snake family").

Ten families of languages are obtained. Their size (interrelationships and relationships with Chomsky families) is investigated in the following sections.

3 Examples and necessary conditions

Example 1. Consider the replicating system

$$\sigma_1 = (\{a, b\}, baba, (a, b)).$$

In all cases, the obtained strings are in $(ba)^+$: we start from $(ba)^2$; when $(ba)^n \rightsquigarrow_g y$, we insert a string $b(ab)^i a, i \geq 0$, in between two symbols ab, hence the string y is of the form $(ba)^m, m > n$.

We have

$$L_g(\sigma_1) = (ba)^+ ba, \text{ for } g \in \{ap, mp, al, ml, af, am\}.$$

Indeed, in any minimal mode (prefix, leftmost, any), we have to insert the string ba. In all free modes we *can* insert ba. Therefore $(ba)^+ba \subseteq L_g(\sigma_1)$. The converse inclusion has been pointed out above. Moreover.

$$L_g(\sigma_1) = \{(ba)^{2^n} \mid n \geq 1\}, \text{ for } g \in \{Mp, Ml, aM, t\}.$$

The maximal prefix (leftmost string, substring) of the form $z = bz'a$ of a string $(ba)^n$ is $(ba)^n$ itself, hence $x \rightsquigarrow_g y$ for g as above means $y = x^2$, which proves the equality.

Example 2. Consider also the system

$$\sigma_2 = (\{a, b\}, abb, (b, b)).$$

We obtain

$$L_g(\sigma_2) = \begin{cases} ab^+b, & \text{for } g \in \{al,\ ml,\ af,\ am\}, \\ abb, & \text{for } g \in \{ap, mp, Mp, t\}, \\ \{ab^{2^n} \mid n \geq 1\}, & \text{for } g \in \{aM,\ Ml\}. \end{cases}$$

In order to obtain counterexamples, we consider a series of necessary condi-
tions.

Lemma 1. *Any snake language is either a singleton or infinite.*

Proof. Consider a replicating system $\sigma = (V, w, (a, b))$. If w cannot be replicated (either no substring ab is present in w, or w contains no suitable substring $z = bz' = z''a$), then $L_g(\sigma) = \{w\}$.

If we have $w \rightsquigarrow_g w'$, then $w = w_1abw_2$, there is $z = bz' = z''a$ in $Sub(w)$ fulfilling the condition associated with g, and $w' = w_1azbw_2$.

Clearly, w' contains again the substring ab, as well as substrings starting with b and ending with a (z is such an example), therefore we can replicate again. This is obvious for all modes $g \in \{al,\ ml,\ Ml,\ af,\ am,\ aM\}$.

If $z \in Pref(w)$, hence $w = zw_3$, then $w' = w_1azbw_2$, and there is $x \in Pref(w')$ such that $x = bx' = x''a$. Indeed, because $z = bz' \in Pref(w)$ and $w_1a \in Pref(w)$, we have $b \in Pref(w_1a) \subseteq Pref(w')$; because $z = z''a$ and $w' = w_1az''abw_2$, the prefix $w_1az''a$ ends with a. Consequently, we can replicate again also in any of the modes $g \in \{ap,\ mp,\ Mp\}$.

For the t case, we have $w = z$ and $w' = w_1awbw_2$. As above, we obtain $w' = bw''a$, hence we can replicate again in this case, too.

The process can be iterated in all modes, hence we obtain an infinite language. $\square$

Lemma 2. *If $L \in SF(g)$, for $g \in \{ap,\ mp,\ Mp,\ t\}$, is an infinite language, then for $c \in Pref(L)$, for each $n \geq 1$ there is $x \in L$ such that $|x|_c \geq n$.*

Proof. Take $\sigma = (V, w, (a, b))$ with $L_g(\sigma) = L$ infinite, for g as specified above. Because L is infinite, there is w' such that $w \rightsquigarrow_g w'$ and the replication can continue an arbitrarily large number of times, at each step using a prefix of the current string (the whole string in the case of $g = t$). All strings in L start with the same symbol, namely with b (from the previous proof, we know that the first symbol is not modified by replication). At each replication, a further copy of the symbol b is introduced: if $x \rightsquigarrow_g y, x = x_1abx_2$, then $y = x_1azbx_2$ for $z \in Pref(x), z = bz'$. Therefore, an arbitrarily large number of occurrences of b can be produced. $\square$

Lemma 3. *If $L \in SF(t)$ is an infinite language, then for both $c \in Pref(L)$ and $d \in Suf(L)$, for each $n \geq 1$ there is $x \in L$ such that $|x|_c \geq n, |x|_d \geq n$.*

Proof. We proceed exactly as in the previous proof, taking into consideration both the first symbol of w and the last symbol of w (they are preserved on these positions by replication, and the number of their occurrences in the current string increases continuously). $\square$

Lemma 4. *Any infinite language $L \in SF(t)$, with $V = alph(L)$, has $\Psi_V(L) = \{\pi \cdot 2^n \mid n \geq 0\}$, for some $\pi \in \mathbf{N}^k, k = card(V)$.*

Poof. If $L = L_t(\sigma)$ for some $\sigma = (V, w, (a, b))$ and L is infinite, then there are arbitrarily long replication chains

$$\rho : w = w_1 \rightsquigarrow_t w_2 \rightsquigarrow_t w_3 \rightsquigarrow_t \dots$$

Clearly, $\Psi_V(w_{i+1}) = 2\Psi_V(w_i), i \geq 1$, hence $\Psi_V(w_i) = 2^{i-1}\Psi_V(w_1), i \geq 1$, and this is true for all chains ρ. Consequently, $\Psi_V(L) = \{\Psi_V(w) \cdot 2^n \mid n \geq 0\}$, that is π in the lemma is $\Psi_V(w)$. $\square$

Of course, Lemma 4 implies Lemma 3.

Lemma 5. *For any infinite language $L \in SF(g)$, $g \in \{Mp, Ml, aM\}$, with $V = alph(L)$, there are $\pi_1, \pi_2 \in \mathbf{N}^k, k = card(V)$, such that $\Psi_V(L) = \{\pi_1 + \pi_2 \cdot 2^n \mid n \geq 0\}$.*

Proof. Take $\sigma = (V, w, (a, b))$ such that $L = L_g(\sigma)$, and examine an arbitrarily long replication chain

$$\rho : w = w_1 \rightsquigarrow_g w_2 \rightsquigarrow_g w_3 \rightsquigarrow_g \dots$$

Assume that $w_1 = x_1abx_2, w_2 = x_1az_1bx_2$, for z_1 maximal in w_1 of the form $z_1 = bz_1' = z_1''a$. This means that when writing $w_1 = y_1z_1y_2$, z_1 starts with the leftmost occurrence of b in w_1 and ends with the rightmost occurrence of a in w_1.

If $w_1 = y_1by_1'$ and this is the leftmost occurrence of b in w_1, then $w_2 = y_1by_1''$ again: if the insertion place is to the right hand of this occurrence of b, then the prefix y_1b of w_1 appears unmodified in w_2; if the insertion place is to the left hand of this occurrence of b, then it must imply this b (being the leftmost), hence y_1 ends with a and we get $w_2 = y_1z_1bx_2 = y_1bz_1'bx_2$, that is again the prefix y_1b is preserved.

Similarly, if a in $w_1 = y_2'ay_2$ is the rightmost occurrence of a in w_1, then the suffix ay_2 appears in w_2, too.

Therefore, $w_1 = y_1z_1y_2, w_2 = y_1z_2y_2$, where z_2 starts with b, ends with a, and it is maximal in w_2 with these properties. Moreover,

$$\begin{aligned} |z_2| &= |w_2| - |y_1y_2| = |w_1z_1| - |y_1y_2| = \\ &= |y_1z_1y_2| + |z_1| - |y_1y_2| = 2|z_1|. \end{aligned}$$

Consequently, at every step of ρ we get a string of the form $y_1z_iy_2$ with $\Psi_V(z_{i+1}) = 2\Psi_V(z_i), i \geq 1$. For $\pi_1 = \Psi_V(y_1y_2)$ and $\pi_2 = \Psi_V(z_1)$, we have the assertion in the lemma (clearly, if $z_i \in Pref(w_i)$, then $z_{i+1} \in Pref(w_{i+1})$).$\square$

In contrast to the last two lemmas, we have

Lemma 6. *For all $g \in \{af,\ am,\ ap,\ mp,\ al,\ ml\}$, if $L \in SF(g)$ is an infinite language, then there is a constant k such that if $x \in L$, then there is $y \in L$ with $|x| < |y| \le |x| + k$.*

Proof. Take $\sigma = (V, w, (a, b))$ and $z \in Sub(w)$ such that $w \rightsquigarrow_g w'$, with $w = w_1abw_2$, $w' = w_1azbw_2$. For $g = af$ we can insert z again. For $g = am$ we can insert z again if z is minimum, or a shorter string otherwise.

If $z \in Pref(w)$, then $z \in Pref(w_1az) \subseteq Pref(w')$, hence for $g = ap$ we can use z again, whereas for $g = mp$ we can use z or a shorter prefix.

The same assertions hold for the leftmost cases of choosing the string to be inserted.

The argument can be iterated, hence at every step $x \rightsquigarrow_g y$ with $w \rightsquigarrow_g^* x$, we have $|x| < |y| \le |x| + |w|$. Taking $k = |w|$, we have the lemma. □

We shall use these lemmas in the next sections when investigating the families $SF(g)$. The reader can easily find languages not satisfying these necessary conditions.

4 The place in Chomsky hierarchy

From Example 1 (each family $SF(g)$ contains infinite languages) and Lemma 1 (there are finite languages not in $SF(g)$, for any $g \in D$), we get

Theorem 1. *The family FIN is incomparable with each family $SF(g), g \in D$.*

From Example 1 we also know that all families $SF(g), g \in \{Mp,\ Ml,\ aM,\ t\}$, contains non-context-free languages. In fact, a stronger result is true.

Theorem 2. *Each family $SF(g), g \in \{Mp,\ Ml,\ aM,\ t\}$, is incomparable with each subfamily of MAT containing at least a non-sigleton finite language.*

Proof. For the system $\sigma = (\{a\}, aa, (a, a))$ we obtain

$$L_g(\sigma) = \{a^{2^n} \mid n \ge 1\},$$

for each $g \in \{Mp,\ Ml,\ aM,\ t\}$. According to [3], this language is not in the family MAT.

On the other hand, there are finite languages not in $SF(g)$, for all $g \in D$. □

Part of the other snake families contain only regular languages:

Theorem 3. $SF(g) \subseteq REG,\ g \in \{am,\ mp\}$.

Proof. Take a replicating system $\sigma = (V, w, (a, b))$. If

$$w = a_1 a_2 \dots a_n,$$

with $a_i \in V, 1 \leq i \leq n, n \geq 1$, then we consider the string

$$w' = s_0 a_1 s_1 a_2 \dots s_{n-1} a_n s_n,$$

where $s_0, s_1, \dots, s_n$ are new symbols.

Identify in w all the positions of the substring ab. Namely, let $1 \leq i_1 < i_2 < \dots < i_r < n$ be these positions:

$$a_{i_q} = a,\ a_{i_q+1} = b,\ 1 \leq q \leq r.$$

All these positions are insertion places.

Identify now by pairs $(j_1, k_1), \dots, (j_t, k_t)$ all minimal substrings of w of the form $z_l = bz_l' = z_l''a$. Namely,

$$z_l = a_{j_l} a_{j_l+1} \dots a_{k_l},\ 1 \leq l \leq t.$$

For each such string z_l and for each insertion position i_q specified above, consider the string

$$z_{l,q} = s_{j_l-1}^{(l,q)} a_{j_l} s_{j_l}^{(l,q)} a_{j_l+1} \dots s_{k_l-1}^{(l,q)} a_{k_l} s_{k_l}^{(l,q)},$$

where $s_h^{(l,q)}$ are new symbols, associated with h, z_l and i_q.

Denote by K the set of all symbols $s_i, 0 \leq i \leq n,\ s_h^{(l,q)}, 1 \leq l \leq t, j_l - 1 \leq h \leq k_l,\ 1 \leq q \leq r$. Construct the finite automaton (with λ-moves)

$$A = (K, V, s_0, \{s_n\}, \delta),$$

where δ is defined as suggested in figure 2, such that it covers all links in w', in strings z_l, as well as the links asked for by insertions of strings z_l in w' on the corresponding positions:

$$\begin{aligned}
&s_i = \delta(s_{i-1}, a_i),\ \text{for } 1 \leq i \leq n,\\
&s_h^{(l,q)} \in \delta(s_{h-1}^{(l,q)}, a_h),\ \text{for } j_l \leq h \leq k_l, 1 \leq l \leq t, 1 \leq q \leq r,\\
&s_{j_l-1}^{(l,q)} \in \delta(s_{i_q}, \lambda),\\
&s_{i_q} \in \delta(s_{k_l}^{(l,q)}, \lambda),\ \text{for } 1 \leq l \leq t, 1 \leq q \leq r.
\end{aligned}$$

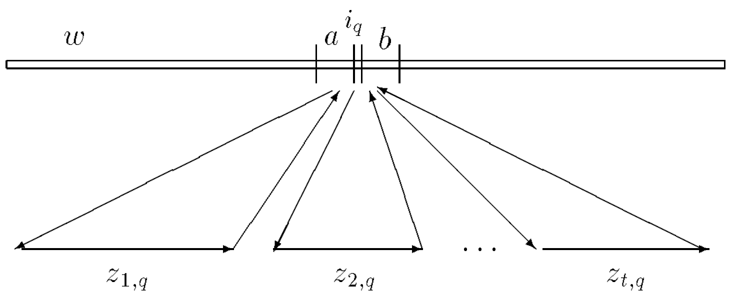

Figure 2

We obtain $L_{am}(\sigma) = L(A)$.

The inclusion $L(A) \subseteq L_{am}(\sigma)$ is obvious: we start from $w \in L_{am}(\sigma)$ and insert, in correct places, correct substrings.

Conversely, consider a substring $x \in L_{am}(\sigma)$. If $x = w$, then $x \in L(A)$. If $w \rightsquigarrow_{am} x$, then $w = w_1abw_2, x = w_1aybw_2$, for some minimal $y \in Sub(w)$, $y = by' = y''a$. From the definition of δ, we have $x \in L(A)$. Being minimal, y does not contain a substring ab, hence the insertion places in x are either insertion places in w_1, w_2, or the two new ones obtained by inserting y, because $ayb = aby'b = ay''ab$. But each of these two subwords ab in ayb involves one symbol already existing in w. If a string $z = bz' = z''a$ is to be inserted in x in one of these places, this is possible also in the automaton A (see again figure 2). Consequently, by induction on the length of the replication chain we find that in each string x such that $w \rightsquigarrow^*_{am} x$, the insertion places correspond to the insertion places in w, hence when $x \rightsquigarrow_{am} x'$, we have $x' \in L(A)$ if $x \in L(A)$. In conclusion, $L_{am}(\sigma) \subseteq L(A)$.

The above discussion covers also the case $g = mp$: then we have $t = 1$ (only one string to be inserted), but the rest of the argument is identical.□

Theorem 4. $SF(g) \subseteq REG,\ g \in \{ml,\ af\}$.

Proof. We use the same construction as in the proof of the previous theorem, with the following differences.

For the case $g = ml$ we have to consider the (unique) minimal leftmost substring z of w of the form $z = bz' = z''a$, instead of the strings $z_1, \ldots, z_t$ considered above. Because the replication mode is minimal leftmost, this string is the only one which can be inserted in any word of $L_{lm}(\sigma)$. (Note that this string z may not be minimal in the am sense: for example, consider $w = cbcbab$, for the insertion context (a, b); in the leftmost minimal mode we have to take $z = bcba$, whereas in the am mode this string is not minimal, because it contains the substring ba. This is the reason why the above proof does not automatically cover the case $g = ml$.)

For the case $g = af$, let $z_1, z_2, \ldots, z_t$ in the proof of the previous theorem be all the substrings of w of the form $z_l = bz'_l = z''_l a$, not containing the substring ab, but with z'_l possibly containing occurrences of b, and also z''_l possibly containing occurrences of a. Let y be any substring of w. Then we can write $y = z_{i_1} z_{i_2} \ldots z_{i_m}$, where z_{i_j} are strings as above. If we want to insert the string y in a position i_q, then from the definition of δ we see that from s_{i_q} we can go to the state in front of each $z_{i_j}, 1 \leq j \leq m$, and back to s_{i_q}, hence we have a path through the states of A covering the string $z_{i_1} z_{i_2} \ldots z_{i_m} = y$. We obtain the inclusion $L_{af}(\sigma) \subseteq L(A)$, also for this case. $\square$

In contrast to the previous theorems, we have

Theorem 5. *The families $SF(ap)$ and $SF(al)$ contain non-context-free languages.*

Proof. Let us consider the replicating system

$$\sigma = (\{a, b, c, d\}, bcabda, (a, b)).$$

In both modes ap and al, all the generated strings start with the symbol b, hence the two modes coincide, $L_{ap}(\sigma) = L_{al}(\sigma)$.

Consider also the regular language

$$R = (bca)^+(bda)^+.$$

Denote $V = \{a, b, c, d\}$ (with this ordering of symbols). If $L_{ap}(\sigma)$ is a context-free language, then the language

$$L = L_{ap}(\sigma) \cap R$$

is context-free, too, hence $\Psi_V(L)$ is a semilinear set. Take the semilinear set

$$T = \{(n, m, p, p) \mid n, m, p \geq 1\}.$$

The family of semilinear sets is closed under intersection. Therefore, $L_{ap}(\sigma) \in CF$ implies the fact that $\Psi_V(L) \cap T$ is a semilinear set.

However, we have

$$\Psi_V(L) \cap T = \{(2^{n+1}, 2^{n+1}, 2^n, 2^n) \mid n \geq 1\}, \qquad (*)$$

which, clearly, is not a semilinear set.

Let us prove the equality $(*)$.

If we have a string $x = (bca)^i(bda)^i$ (initially we have $i = 1$) and we use exactly x as the string to be inserted, namely in between a and b at the middle of x, then we obtain $y = (bca)^{2i}(bda)^{2i}$. The process can be iterated, hence each string of the form $z = (bca)^{2^j}(bda)^{2^j}, j \geq 0$, is in $L_{ap}(\sigma)$. Such strings are also in R, hence they are in L. As $\Psi_V(z) = (2^{j+1}, 2^{j+1}, 2^j, 2^j)$, we have the inclusion $\supseteq$.

Conversely, let us examine the strings $y \in L$ for which $\Psi_V(y) \in T$. Being in L, y is of the form $y = (bca)^i(bda)^j, i, j \geq 1$. Because $\Psi_V(y) \in T$, we must have $i = j$. Therefore we have to consider the strings in $L_{ap}(\sigma)$ of the form $y = (bca)^i(bda)^i, i \geq 1$.

Because we work in the *ap* mode, when replicating a string $(bca)^k(bda)^k$, $k \geq 1$, we have to choose a string of the form $z_1 = (bca)^r, r \leq k$, or of the form $z_2 = (bca)^k(bda)^s, s \leq k$. Therefore, in every moment, what we obtain is a string y' with $|y'|_c \geq |y'|_d$. If we get a string y' with $|y'|_c > |y'|_d$, then no string y'' such that $y' \rightsquigarrow^*_{ap} y''$ will have $|y''|_c = |y''|_d$.

If we have a string y' which is not an element of $(bca)^+(bda)^+$, that is a string of the form $u_1du_2cu_3$, then no string y'' such that $y' \rightsquigarrow^*_{ap} y''$ will be in $(bca)^+(bda)^+$.

Consequently, in order to generate strings of the form $y = (bca)^i(bda)^i$ we have to use only strings of the same form, and at every replication step the inserted string is the whole current string. This implies that each replication doubles the whole string, hence i above is of the form $i = 2^n, n \geq 0$. This proves the inclusion $\subseteq$ in relation $(*)$. □

Corollary. *Each family $SF(ap)$, $SF(al)$ is incomparable with each subfamily of CF containing at least a non-singleton finite language.*

Because MAT contains non-semilinear languages [1], we cannot infer from the previous theorem that $SF(ap)$ and $SF(al)$ contain languages not in the family MAT, as in Theorem 2.

The difference between the modes *ap, al* and the modes *af, am, ml, mp* is quite non-expected (*af* and *am* behave in the same way, but *ap* and *mp*, or *al* and *ml* do not).

Because $x \rightsquigarrow_g y$ implies $|x| < |y|$, for all $g \in D$, it is easy to prove that

Theorem 6. *$SF(g) \subseteq CS$, for all $g \in D$.*

5 Comparing snake families

From Lemmas 4 and 5 we find that for the system σ_1 in Example 1, the languages $L_g(\sigma_1)$, for $g \in \{ap,\ mp,\ al,\ ml,\ af,\ am\}$, are not in $SF(g')$, for $g' \in \{Mp,\ Ml,\ aM,\ t\}$, hence

Lemma 7. *$SF(g) - SF(g') \neq \emptyset$, for all $g \in \{ap,\ mp,\ al,\ ml,\ af,\ am\}$, $g' \in \{Mp,\ Ml,\ aM,\ t\}$.*

From Lemma 6 we find that $L_g(\sigma_1) \notin SF(g')$, for $g \in \{Mp,\ Ml,\ aM, t\}$, $f \in \{af,\ am,\ al,\ ml,\ ap,\ mp\}$, hence

Lemma 8. *$SF(g) - SF(g') \neq \emptyset$ for all $g \in \{Mp,\ Ml,\ aM,\ t\}$, $f \in \{af,\ am,\ al,\ ml,\ ap,\ mp\}$.*

Combining these lemmas, we get

Theorem 7. *Each family $SF(g), g \in \{Mp,\ Ml,\ aM,\ t\}$, is incomparable with each family $SF(g'), g' \in \{af,\ am,\ al,\ ml,\ ap,\ mp\}$.*

From Example 2 and Lemma 2, we obtain

Lemma 9. *$SF(g) - SF(g') \neq \emptyset$, for all $g \in \{al,\ ml,\ af,\ am,\ aM,\ Ml\}$, $g' \in \{ap,\ mp,\ Mp,\ t\}$.*

Theorem 8. *$SF(Ml) = SF(aM)$.*

Proof. Take $\sigma = (V, w, (ab))$ and a replication step $z \rightsquigarrow_{aM} z'$. We have $z = z_1abz_2, z' = z_1axbz_2$, for $x \in Sub(z), z = x_1xx_2,\ x = bx' = x''a$, and x is maximal in z, no superstring y of x can be written as $y = by' = y''a$. If x is not leftmost in z, then $z = x'_1bx'_2$ such that $|x'_1| < |x_1|$. Then x is not maximal: $z = x'_1yx_2$ for $y = by' = y''a, |y| > |x|$, hence $x \in Sub(y)$, a contradiction. Therefore $z \rightsquigarrow_{Ml} z'$, too, that is $L_{aM}(\sigma) \subseteq L_{Ml}(\sigma)$.

Conversely, it is easy to see that $z \rightsquigarrow_{Ml} z'$ implies $z \rightsquigarrow_{aM} z'$, hence $L_{Ml}(\sigma) \subseteq L_{aM}(\sigma)$. In conclusion, $L_{Ml}(\sigma) = L_{aM}(\sigma)$, hence $SF(aM) \subseteq SF(Ml)$.

The converse inclusion can be obtained in the same way, and this completes the proof. □

Theorem 9. *$SF(t) \subset SF(g),\ g \in \{Ml,\ Mp,\ aM\}$.*

Proof. Take a replicating system $\sigma = (V, w, (a, b))$. If $L_t(\sigma) = \{w\}$, then $L_t(\sigma) = L_g(\sigma')$, for $\sigma' = (V \cup \{c\}, w, (c, c))$, with $c \notin V$, for all g as above.

If $L_t(\sigma) \neq \{w\}$, then $w = bw'a$ and all z such that $w \rightsquigarrow_g^* z$ are of the form $z = bz'a$ for all g. Consequently, at every moment of a replication chain the obtained string is at the same time the maximal prefix, maximal leftmost, and arbitrary maximal of the form bya. Therefore, $L_t(\sigma) = L_g(\sigma)$ for all $g \in \{Ml,\ Mp,\ aM\}$. We obtain $SF(t) \subseteq SF(g)$, for g as in the theorem.

From Lemma 7 we know that these inclusions are proper for $g \in \{aM, Ml\}$. For $g = Mp$ we consider the system

$$\sigma = (\{a, b\}, aab, (a, a)).$$

We obtain

$$L_{Mp}(\sigma) = \{a^{2^n} b \mid n \geq 1\}.$$

According to Lemmas 3 and 4, this language is not in the family $SF(t)$. $\square$

Lemma 10. *$SF(g) - SF(g') \neq \emptyset$, $g \in \{am,\ ml,\ mp\}$, $g' \in \{aM,\ t,\ Mp,$ $ap,\ al,\ Ml,\ af\}$.*

Proof. Consider the replicating system

$$\sigma = (\{a, b\}, aabaa, (a, a)).$$

For g as above, the string to be inserted is always a, hence

$$L_g(\sigma) = a^+ aba^+ a.$$

Take a system $\sigma' = (V, aabaa, (c, d))$ such that $L_g(\sigma) = L_{g'}(\sigma')$, for any g' as above. (Always the shortest string in a language is the axiom of the replicating system generating it.) Clearly, $c, d \in \{a, b\}$. If one of c, d is equal with b, then in each $z \rightsquigarrow_{g'} z'$ one introduces one new occurrence of b, which leads to parasitic strings. Therefore, $(c, d) = (a, a)$. For all modes g', we can choose for insertion the whole current string (this is mandatory for $g' \in \{aM,$ $t,\ Mp,\ Ml\}$) and this again doubles the number of occurrences of the symbol b. The equality $L_g(\sigma) = L_{g'}(\sigma')$ is not possible. $\square$

Theorem 10. *The family $SF(mp)$ is incomparable with each family $SF(g'), g' \in \{al,\ af\}$.*

Lemma 11. *$SF(g) - SF(g') \neq \emptyset$, for all $g \in \{af,\ ap,\ al\}$, $g' \in \{am,\ mp,$ $ml\}$.*

Proof. Consider again the system

$$\sigma = (\{a, b\}, aabaa, (a, a)).$$

For all modes g as above, we can replicate

$$aabaa \rightsquigarrow_g a(aabaa)abaa = z$$

(we insert the whole string, in between the first two occurrences of a).

Assume that $L_g(\sigma) = L_{g'}(\sigma')$ for some $\sigma' = (V, aabaa, (c, d))$. If $(c, d) =$ (a, a), then we have to insert a (this is the minimal substring of the form $x = ax' = x''a$), hence the number of occurrences of b is not increased, the string z above cannot be obtained. If $(c, d) = (a, b)$, then the string to be inserted is ba, hence we obtain

$$L_{g'}(\sigma') = a(ab)^+ aa.$$

If $(c, d) = (b, a)$, then the string to be inserted is ab, and we obtain

$$L_{g'}(\sigma') = aa(ba)^+a.$$

In none of the possible cases we can obtain the string z above, hence the equality $L_g(\sigma) = L_{g'}(\sigma')$ is not possible. $\square$

Theorem 11. *Each family $SF(g), g \in \{af,\ ap,\ al\}$, is incomparable with each family $SF(g')$, $g' \in \{am,\ mp,\ ml\}$.*

Theorem 12. $SF(Mp) \subset SF(lM)$.

Proof. If $\sigma = (V, w, (a, b))$ is a replicating system and $L_{Mp}(\sigma) = \{w\}$, then we obviously have $L_{Mp}(\sigma) \in SF(lM)$.

If $L_{Mp}(\sigma)$ is infinite, then w is of the form $w = zw_1, z = bz' = z''a$, and all strings in $L_{Mp}(\sigma)$ start with the symbol b. Consequently, the leftmost string to be inserted is always a prefix of the current string, hence $L_{Mp}(\sigma) = L_{lM}(\sigma)$, that is $SF(Mp) \subseteq SF(lM)$.

This inclusion is proper, because $L = \{ba^{2^n} \mid n \geq 1\}$ is not in the family $SF(Mp)$ (Lemma 2), but $L = L_{lM}(\sigma)$, for $\sigma = (\{a, b\}, baa, (a, a))$. $\square$

Theorem 13. $SF(mp) \subset SF(ml),\ SF(ap) \subset SF(al)$.

Proof. The inclusions follow as in the previous proof: if $\sigma = (V, w, (a, b))$, then $L_{mp}(\sigma) = L_{ml}(\sigma)$ and $L_{ap}(\sigma) = L_{al}(\sigma)$.

The strictness follows from Lemma 9. $\square$

Lemma 12. $SF(g) - SF(am) \neq \emptyset,\ g \in \{mp,\ ml\}$.

Proof. Consider the system

$$\sigma = (\{a, b, c\}, bcabacbca, (a, b)).$$

For both $g = mp$ and $g = ml$, the string to be inserted is always bca. This is obvious for the axiom of σ, $bcabacbca$. We get

$$bcabacbca \rightsquigarrow_g (bca)^2bacbca,$$

hence after n insertions we generate the string $(bca)^n bacbca$. Therefore, for each string in $L_g(\sigma)$ we have $|x|_a = |x|_b = |x|_c$.

Assume now that $L_g(\sigma) = L_{am}(\sigma')$ for some $\sigma' = (V, w, (d, e))$. Being the shortest string of $L_g(\sigma)$, $bcabacbca = w$ is the axiom of σ'.

Assume that $d \neq e$. Whichever d, e are among a, b, c, because all substrings ed appear in w, we can choose such a minimal substring to be inserted, and we can obtain a string w' such that $|w'|_d = |w'|_e = |w'|_f + 1$, for $\{a, b, c\} = \{d, e, f\}$. Such a string is not in $L_g(\sigma)$, a contradiction. If $d = e$, then the string to be inserted is $d = e$, which increases only the number of occurrences of $d = e$, again leading to parasitic strings. $\square$

Lemma 13. $SF(g) - SF(af) \neq \emptyset,\ g \in \{ap,\ al\}$.

Proof. Take again the system

$$\sigma = (\{a, b, c\}, bcabacbca, (a, b)),$$

in the previous proof. For both $g = ap$ and $g = al$, the string to be inserted starts with bca, the prefix of the axiom. Moreover, in any moment of a replication chain, the prefix bca is preserved, hence it must be considered for insertion. Consequently, there is no constant k and there are no symbols $d, e \in \{a, b, c\}$ such that $L_g(\sigma)$ contains strings $w_1, w_2, \ldots$, such that $|w_i|_d$ is strictly increasing with i and $|w_i|_e \leq k$.

Assume now that $L_g(\sigma) = L_{af}(\sigma')$, for some $\sigma' = (V, w, (d, e))$. Clearly, $w = bcabacbca$. Irrespective which the symbols d, e are, the substring ed appears in w (and in all strings produced by replication). Therefore, $L_{af}(\sigma')$ contains strings with arbitrarily large number of occurrences of d, e and a bounded number of occurrences of $f = \{a, b, c\} - \{d, e\}$. Such strings are not in $L_g(\sigma)$, a contradiction. $\square$

Theorem 14. *The families in the next pairs are incomparable:* $(SF(mp), SF(am))$, $(SF(ap), SF(af))$.

Lemma 14. $SF(af) - SF(al) \neq \emptyset,\ SF(am) - SF(ml) \neq \emptyset$.

Proof. Consider the system $\sigma = (\{a, b, c\}, w, (a, b))$ with $w = bcacbab$. In both modes *af* and *am* of choosing the string to be inserted, we can choose the substring ba of w, hence we can obtain strings containing arbitrarily many occurrences of a and b, but only a bounded number of occurrences of c. In order to generate such strings by a system $\sigma' = (V, w, (d, e))$ in one of the modes *al, ml*, we must have $\{d, e\} = \{a, b\}$. If $(d, e) = (a, b)$, then the string to be inserted will start with the prefix bca. If $(d, e) = (b, a)$, then the string to be inserted starts at the leftmost occurrence of a (on the third position of each generated string) and contains the neighbouring occurrence of c. In both cases, the number of a, b, c occurrences increases simultaneously, hence we cannot generate the above mentioned strings of $L_{af}(\sigma), L_{am}(\sigma)$. $\square$

Theorem 15. *The families in the next pairs are incomparable*: $(SF(af), SF(al))$, $(SF(am), SF(ml))$.

Proof. Combine Lemma 14 with Lemma 13 (for the pair (*af, al*)) and Lemma 14 with Lemma 12 (for the pair (*am, ml*)). $\square$

Consequently, the relations between all families $SF(g), g \in D$, are settled: excepting the following strict inclusions and equality, all two families are incomparable:

$$SF(mp) \subset SF(ml),\ SF(ap) \subset SF(al),$$
$$SF(t) \subset SF(Mp) \subset SF(Ml) = SF(aM).$$

6 Variants to be considered

We have mentioned already in Section 2 that we can consider various restrictions on the place where the insertion is performed. Here we have investigated the *free* mode (condition (1) in Definition 2), but we can also restrict the insertion to the *leftmost* place where it is possible. When we have only one insertion context, as we have dealt with here, this position is unambiguously specified. When we have several insertion contexts, then we can consider both *locally leftmost* positions (choose an insertion context (u, v) and look for the leftmost occurrence of uv in the current string), or the *globally leftmost* (the leftmost substring uv, for all insertion contexts (u, v)).

Thus, the general form of a replicating system is

$$\sigma = (V, w, C),$$

where V is an alphabet, $w \in V^*$ and $C \subseteq V^+ \times V^+$ is a finite set of insertion contexts. For the three modes of choosing the place of insertion (free $= f$, locally leftmost $= ll$, and globally leftmost $= gl$) and the ten modes of choosing the string to be inserted, identified by the elements of D as above, we get thirty modes of replication, hence thirty families of languages, $SF(g_1, g_2), g_1 \in \{f, ll, gl\}, g_2 \in D$.

Most of the proofs of this paper are not valid for this general case. In particular, the regularity of languages in families $SF(g_1, g_2)$, for $g_1 \in \{f, ll, gl\}$, $g_2 \in \{mp, ml, af, mf\}$, is *open* (and it seems to be a non-trivial problem).

If we distinguish between free (f) and leftmost (l) insertion places also for the simple replicating systems (for $\sigma = (V, w, (a, b))$, the ll and the gl modes coincide), then all the results in the previous sections are true for both the f and the l modes, including Theorems 3, 4, with a slight modification of the proof (we have to consider only one insertion place, i_1). However, no comparison between families $SF(f, g)$ and $SF(l, g)$ is done.

Of course, not only the generative capacity of replicating systems is to be investigated. Closure, decidability, and descriptional complexity problems are among the usual ones in language theory, and they look quite natural also in this case. (For instance, does the number of insertion context, or the maximal length of their elements, induce an infinite hierarchy of languages ?) Moreover, because in several cases, the strings of the generated languages are produced in a sequence (this is the case for simple replicating systems inserting in the leftmost place a string choosen in one of the modes *t, mp, Mp, ml, Ml*), it makes sense to consider the *growth function* of such a (snake) language.

The snake replicating machineries deserve further investigations.

References

[1] J. Dassow, Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, Heidelberg, 1989.

[2] B. S. Galiukschov, Semicontextual grammars, *Mat. logica i mat. ling.*, Kalinin Univ., 1981, 38 – 50 (in Russ.).

[3] D. Hauschild, M. Jantzen, Petri nets algorithms in the theory of matrix grammars, *Acta Informatica*, 31 (1994), 719 – 728.

[4] T. Head, Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors, *Bull. Math. Biology*, 49 (1987), 737 – 759.

[5] T. Head, Gh. Păun, D. Pixton, Languages and DNA recombination, in *The Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.), in preparation.

[6] L. Kari, *Insertion and Deletion in Formal Languages*, PhD Thesis, Turku University, 1992.

[7] S. Marcus, Contextual grammars, *Rev. Roum. Math. Pures Appl.*, 14 (1969), 1525 – 1534.

[8] Gh. Păun, G. Rozenberg, Contextual grammars, in *The Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.), in preparation.

[9] A. Salomaa, *Formal Languages*, Academic Press, New York, 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**