Kybernetika

Miroslav Langer

On hierarchy of the positioned eco-grammar systems

Kybernetika, Vol. 50 (2014), No. 5, 696–705

Persistent URL: `http://dml.cz/dmlcz/144101`

KYBERNETIKA — VOLUME *50* (2014), NUMBER 5, PAGES 696–705

ON HIERARCHY OF THE POSITIONED ECO-GRAMMAR SYSTEMS

Miroslav Langer

Positioned eco-grammar systems (PEG systems, for short) were introduced in our previous papers. In this paper we engage in a new field of research, the hierarchy of PEG systems, namely in the hierarchy of the PEG systems according to the number of agents presented in the environment and according to the number of types of agents in the system.

Keywords: positioned eco-grammar systems, hierarchy, eco-grammar systems

Classification: 22E46, 53C35, 57S20

1. INTRODUCTION

In the last decade of the 20th century a systematic research of the collective behaviour of formal grammars began. In [1] basic models of the grammar system were introduced. One of the models based on this paper are the eco-grammar systems (see [2]).

Based on the eco-grammar systems, we have introduced positioned eco-grammar systems; the PEG systems for short (see [7]). Similarly as in eco-grammar systems, the motivation is to describe the interplay between an evolving environment and the community of agents living in this environment, whereas we focus on agent's position in the environment. We combine the approaches from the PM-colonies (see [11]) the and eco-grammar systems (see [2, 3]). The environment of the PEG system is represented by a 0L scheme (see [4]) and the position of each agent in the environment is given by its identifier.

In this paper we describe the hierarchy of the positioned eco-grammar systems based on the number of agents present in the environment and the number of types of agents in the system. The results obtained for the hierarchy of the original eco-grammars systems and it's variants can be found e. g. in [6, 13].

The paper is structured as follows: in the section 2 we give the basic notations and definitions of the positioned eco-grammar systems. In the section 3 we show that the generative power of the positioned eco-grammar systems depends on the number of types of agents in the system, and on the number of agents present in the environment as well.

DOI: 10.14736/kyb-2014-5-0696

2. POSITIONED ECO-GRAMMAR SYSTEMS

In this section we present basic definitions of the positioned eco-grammar systems. The positioned eco-grammar systems combine the approach of the eco-grammar systems (for more information see e. g. [2, 3]) and the PM colonies (see [11]). Similarly as eco-grammar systems, each positioned eco-grammar system consists of a developing environment and collection of agents (organisms). Moreover position of each agent in the environment is fixed in accordance with PM colonies. An agent can affect the environment only in its neighbouring left or right position and other symbols in the environment are rewritten in totally parallel manner like in L systems [12].

We will follow the definitions with very simple example to explain it.

Definition 2.1. A positioned eco-grammar system (PEG system, for short) of degree m, $m \geq 1$, is an $(m+3)$-tuple $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$, where

- V_E is a finite nonempty alphabet of the environment,
- $N_B = \{[j] : 1 \leq j \leq m\}$ is a set of identifiers of the agents, $[j]$ defines position of the jth type agent in the environment,
- $E = (V_E, P_E)$ is a $0L$ scheme – the environment,
- $B_j = ([j], Q_j)$, is the jth type agent for $1 \leq j \leq m$ and Q_j is a set of rules of the form:
  - $a[j] \to u$ or $[j]b \to v$ where $a, b \in V_E$ is symbol marking vicinity with agent, $u, v \in (V_E \cup N_B)^*$.
    In what follows we will use for rules description also $a[j]b \to u$, where $ab \in V_E$. So in every rule either $b = \varepsilon$ or $a = \varepsilon$ but not both.

Note that a PEG system requires at least one agent by the definition.

Example 2.2. PEG system $\Sigma = (V_E, N_B, E, B_1)$, where:

- $V_E = (\{a, b, c\}$,
- $N_B = \{[1]\}$,
- $E = (V_E, \{a \to a, b \to b, c \to c\})$,
- $B_1 = ([1], \{[1]c \to \varepsilon, [1]c \to a[1]ca, [1]c \to b[1]cb\})$.

We consider PEG system with one type of the agent B_1 and non-evolving environment. The growth of environment (generating the language) ensures the agent.

A configuration of the PEG system determines the actual state of a PEG system, it means the actual string of the environment together with collection of agents with fixed positions in the environment and it is introduced as follows:

Definition 2.3. A configuration of the positioned eco-grammar system $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ is a string v, where $v \in (V_E \cup N_B)^*$. The starting configuration is called an axiom.

Example 2.4. Let the axiom of the system Σ be $w = [1]c$.

We describe the derivation step in the PEG system as follows:

Definition 2.5. A derivation step of the positioned eco-grammar system $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ is a binary relation $\Longrightarrow_\Sigma$ on $(V_E \cup N_B)^*$, such that $w \Longrightarrow_\Sigma w'$ iff

- $w = \alpha_0 a_1 [j_1] b_1 \alpha_1 \ldots \alpha_{n-1} a_n [j_n] b_n \alpha_n$, where
  $\alpha_k \in V_E^*$ for $0 \leq k \leq n$ and $a_k b_k \in V_E, [j_k] \in N_B, 1 \leq k \leq n$,
- $w' = \alpha_0' \beta_1 \alpha_1' \ldots \alpha_{n-1}' \beta_n \alpha_n'$, where
  $a_k[j_k]b_k \rightarrow \beta_k \in Q_k$ for $1 \leq k \leq n$ and $\alpha_k \Rightarrow_E \alpha_k'$ for $0 \leq k \leq n$.

By $\Longrightarrow^*$ we denote the reflexive and transitive closure of the relation $\Longrightarrow$.

Example 2.6. Let us show first few derivation steps of the PEG system Σ:
$[1]c \Rightarrow a[1]ca \Rightarrow aa[1]caa \Rightarrow aab[1]cbaa \Rightarrow aaba[1]cabaa \ldots$

Agents in the PEG system work parallel. Each agent rewrites one symbol on its right or left hand side together with its own identifier, in each derivation step. The rest of the symbols (i. e. those not touched by agents) are rewritten by the rules of the environment in totally parallel manner like in Lindenmayer system. Each agent has to rewrite its own symbol otherwise the derivation is blocked. This is in the cases when there is no rule for the agent with its actual right or left context, or when agent has no free context to rewrite.

The language defined by the positioned eco-grammar system is given by all words produced by the system from the axiom, ignoring agents identifiers.

Definition 2.7. The language defined by the positioned eco-grammar system $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ and the axiom $w, w \in (V_E \cup N_B)^+$ is a set of strings:

$$L(\Sigma, w) = \{\gamma(u) : w \Rightarrow_\Sigma^* u, u \in (V_E \cup N_B)^*\},$$

where γ is the morphism such that $\gamma(a) = a$ for $a \in V_E$ and $\gamma(b) = \varepsilon$ for $b \in N_B$.

Example 2.8. As the reader can easily verify, the language defined by PEG system Σ is $L(\Sigma, w) = \{wcw^R : w \in \{a, b\}^*\}$.

The class of languages defined by the positioned eco-grammar systems (PEG languages) is denoted as $\mathcal{L}(\text{PEG})$.

Following example shows how the environment can exert influence up the agents and vice versa. We will consider female agent B_F, male agent B_M and child B_C. Male agent builds the nest (n) where female agent lays an egg (e) which is inseminate by the male agent. The female agent broods from the inseminated egg (i) child agent. Before the male can agent build the nest he must wait for the grass to grow from small to grown (in the environment is used rule $s \rightarrow g$). This is how the environment can influence the behaviour of the agents. The environment cannot grow (increase number of the symbol b) before the child agent changes symbols a to the b. In this way the agents can affect behaviour of the environment.

Example 2.9. PEG system $\Sigma = (V_E, N_B, E, B_F, B_M, B_C)$, where:

- $V_E = \{a, b, s, g, e, i\}$,
- $N_B = \{[F], [M], [C]\}$,
- $E = (V_E, \{a \to a, b \to b, b \to bb, s \to s, s \to g, g \to g, n \to n, e \to e, i \to i\})$,
- $B_F = ([F], \{a[F] \to a[F], [F]n \to [F]e, [F]i \to [C]\})$,
- $B_M = ([M], \{[M]a \to [M]a, g[M] \to n[M], e[M] \to i\})$,
- $B_C = ([C], \{a[C] \to b[C], [C]a \to [C]b, b[C] \to [C]b, [C]b \to b[C]\})$.

Let the axiom of the system be $w = a[F]s[M]a$.
One of the possible derivations in the PEG system Σ with axiom w is as follows:

$a[F]s[M]a \Rightarrow a[F]s[M]a \Rightarrow a[F]g[M]a \Rightarrow a[F]n[M]a \Rightarrow a[F]e[M]a \Rightarrow a[F]e[M]a$
$\Rightarrow a[F]ia \Rightarrow a[C]a \Rightarrow b[C]a \Rightarrow bb[C]b \Rightarrow b[C]bbb \Rightarrow [C]bbbbbb \ldots.$

3. HIERARCHY OF THE POSITIONED ECO-GRAMMAR SYSTEMS

We described the generative power of the positioned eco-grammar systems in the context of the well-established grammars and grammar systems in our previous papers (see [5, 8, 9]). In this paper we focus on the question whether it is possible to structure PEG languages based on the structural properties of the system.

In the positioned eco-grammar systems the positions of the agents are given by special symbols. These components in the parallel environment cause local changes in the configurations of the system. Location of the agent in the environment and the context of the agent directly affects the generative power of the PEG system. In this section we classify the PEG languages with respect to the number of (types of) agents in the definition of underlying PEG system and with respect to the maximal number of the agents present in the environment. We show that such classifications determine the infinite hierarchy on the PEG languages in both cases.

Known results concerning the hierarchy of the various types of the eco-grammar systems can be found e. g. in [6, 13].

3.1. Hierarchy based on the number of agents in the environment

This subsection covers the division of the PEG systems according to the number of agents present in the environment. We will consider PEG systems with at most n agents present in the environment in every derivation step, no matter what type the agents are. It means that we count all the identifiers of the agents regardless of type of the agent.

Definition 3.1. A PEG system $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ with axiom $w, w \in (V_E \cup N_B)^+$ is a PEG system with index n, we write $(\Sigma, w) \in \mathrm{PEG}_n$ for short, if $|u|_{N_B} \leq n$ for all $u \in (V_E \cup N_B)^*$ such that $w \Rightarrow_\Sigma^* u$.

A language L is a PEG language with index n, $L \in \mathcal{L}(\mathrm{PEG}_n)$ if there is a PEG_n system Σ with axiom w such that $L = L(\Sigma, w)$ and for no PEG_m system Σ' and w' $m < n$ it holds $L = L(\Sigma', w')$.

Example 3.2. Let us consider the language $L_1 = \{a^i b^i c^i : i \geq 1\}$ and PEG system $\Sigma_1 = (\{a,b,c\}, \{[1]\}, E, B_1)$, where:

- $E = (\{a,b,c\}, \{a \to a, b \to b, c \to c\})$,
- $B_1 = ([1], \{a[1] \to aa[1]b, [1]c \to [1]cc\})$.

Let the axiom of the system be $w_1 = a[1]b[1]c$. Then
$a[1]b[1]c \Rightarrow aa[1]bb[1]cc \Rightarrow^{n-2} a^n[1]b^n[1]c^n \Rightarrow a^n a[1]bb^n[1]cc^n \Rightarrow \ldots$
Therefore $L(\Sigma_1, w_1) = L_1$ and (Σ_1, w_1) is of index 2, i.e. $(\Sigma_1, w_1) \in \mathrm{PEG}_2$. So $L_1 \in \mathcal{L}(\mathrm{PEG}_t)$ for $t \geq 2$.

To prove $L_1 \in \mathcal{L}(\mathrm{PEG}_2)$ we will verify that L_1 does not belong to $\mathcal{L}(\mathrm{PEG}_1)$, i.e. there is no PEG system Σ and axiom w with index 1 such that $L(\Sigma, w) = L_1$.

Assume contrary that L_1 belongs to $\mathcal{L}(\mathrm{PEG}_1)$ and PEG_1 system $\Sigma_0 = (\{a,b,c\}, \{[1]\}, E, B_1, \ldots, B_m)$ with axiom $w_0 = u_0[1]v_0$ generates L_1. Reader can verify that $E = (\{a,b,c\}, P)$ in Σ_0 is deterministic and $P = \{a \to a, b \to b, c \to c\}$. So, all changes in the environment due to the activity of agents.

Let $u[j]v, 1 \leq j \leq m$ be the word derived from the axiom in Σ_0 and $uv = a^t b^t c^t$ for some $t \geq 1$. We assume that in the next derivation step rule $x[j]y \to z$ is used, where $x, y \in \{a,b,c,\varepsilon\}, x = suf(u), y = pref(v)$ and one of the symbols x, y is empty and z contains one position of some agent [i]. For the position of $[j]$ exactly one of the following cases holds:

- $y = a$. Then rule of type $a[j] \to a^r[i]$ or $[j]a \to a^r[i]$ for some $r \neq 1$ has to be used in next derivation step. (Position and type of the agent $[i]$ on the right hand side of the used rule can be arbitrary.) Therefore for $u[j]v \Rightarrow w$ it holds $|w|_a \neq |w|_b$ and $\gamma(w)$ does not belong to L_1.
- $y = b$. Then $u[j]v \Rightarrow w$ gives $|w|_c < |w|_a$ and $|w|_c < |w|_b$. So $\gamma(w)$ does not belong to L_1.
- $y = c$. Then $u[j]v \Rightarrow w$ where $|w|_a < |w|_b$ or $|w|_a < |w|_c$. So $\gamma(w)$ does not determine the word from L_1. In all cases we got the contradiction so the assumption that L_1 belongs to $\mathcal{L}(\mathrm{PEG}_1)$ does not hold and $L_1 \in \mathcal{L}(\mathrm{PEG}_2)$.

These considerations allow us state the following theorem.

Theorem 3.3. The class of PEG languages with respect to the measure index forms an infinite hierarchy: $\mathcal{L}(\mathrm{PEG}_n) \subset \mathcal{L}(\mathrm{PEG}_{n+1})$ for $n \geq 1$ (i.e. proper subsets).

Proof. Obviously $\mathcal{L}(\mathrm{PEG}_n) \subseteq \mathcal{L}(\mathrm{PEG}_{n+1})$ for $n \geq 1$.

Consider language
$$L_{n+1} = \{a_1^i a_2^i \ldots a_{2n}^i a_{2n+1}^i : i \geq 1\}.$$
We will show that $L_{n+1} \in \mathcal{L}(\mathrm{PEG}_{n+1}) - \mathcal{L}(\mathrm{PEG}_n)$.

First we give a PEG system Σ_{n+1} which generates the language L_{n+1} from axiom and it needs $n+1$ occurrences of agent [1] in the environment to generate words in L_{n+1}: $\Sigma_{n+1} = (V_E, \{[1]\}, E, B)$, where

- $V_E = \{a_1, a_2, \dots, a_{2n+1}\}$,
- $E = (V_E, \{a_1 \to a_1, a_2 \to a_2, \dots, a_{2n+1} \to a_{2n+1}\})$,
- $B = (\{[1]\}, \{a_{2k-1}[1] \to a_{2k-1}^2[1]a_{2k} : 1 \le k \le n\} \cup \{[1]a_{2n+1} \to [1]a_{2n+1}^2\})$.

The axiom of the PEG_{n+1} is $w = a_1[1]a_2a_3[1]a_4 \dots a_{2n-1}[1]a_{2n}[1]a_{2n+1}$.

Derivations proceed as follows:
$a_1[1]a_2 \dots a_{2n-1}[1]a_{2n}[1]a_{2n+1} \Rightarrow a_1^2[1]a_2^2 \dots a_{2n-1}^2[1]a_{2n}^2[1]a_{2n+1}^2 \Rightarrow$
$a_1^3[1]a_3^3 \dots a_{2n-1}^3[1]a_{2n}^3[1]a_{2n+1}^3 \Rightarrow a_1^4[1]a_4^4 \dots a_{2n-1}^4[1]a_{2n}^4[1]a_{2n+1}^4 \Rightarrow \dots.$

Hence the $L(\Sigma_{n+1}, w) = \{a_1^i a_2^i \dots a_{2n}^i a_{2n+1}^i : i \ge 1\}$.

Therefore $L(\Sigma_{n+1}, w) = L_{n+1}$ and (Σ_{n+1}, w) is of index $n+1$, i.e. $(\Sigma_{n+1}, w) \in \mathrm{PEG}_{n+1}$. So $L_{n+1} \in \mathcal{L}(\mathrm{PEG}_t)$ for some $t, t \le n+1$.

To prove that $L_{n+1} \in \mathcal{L}(\mathrm{PEG}_{n+1})$ we will verify that $L_{n+1} \notin \mathcal{L}(\mathrm{PEG}_n)$, i.e. there is no PEG system Σ and axiom w with index n such that $L(\Sigma, w) = L_{n+1}$.

Assume contrary that L_{n+1} belongs to $\mathcal{L}(\mathrm{PEG}_n)$ and PEG_n system $\Sigma_n = (V_E, N_B, E, B_1, \dots, B_m)$ with axiom w_n generates L_{n+1}. We verify that $E = (V_E, P)$ in Σ_n is deterministic and $P = \{a_i \to a_i : 1 \le i \le 2n+1\}$.

It follows from the structure of L_{n+1} that for $a_i \to \alpha \in P$ we have $\alpha = a_i^s$ for some s. Assume that P is not deterministic and $a_i \to a_i^j, a_i \to a_i^k \in P$ for $j < k$. Let derivation of $u = a_1^r a_2^r \dots a_{2n}^r a_{2n+1}^r \in L_{n+1}$ for some r uses the rule $a_i \to a_i^j$. Then also the word $v = a_1^r a_2^r \dots a_i^{r+k-j} \dots a_{2n}^r a_{2n+1}^r \in L(\Sigma, w)$ but $v \notin L_{n+1}$.

Evidently for rules $a_i \to a_i^j, 1 \le i \le 2n+1$ in deterministic P it holds $j = 1$. Otherwise all words from L_{n+1} are not in $L(\Sigma_n, w_n)$

So, all the changes in the environment are done by the activity of the agents. Let $u[j]v \to w$ is in B_j. Then from the structure of L_{n+1} it holds that $|alph(w) \cap V_E| \le 2$. This gives that for $x \Rightarrow y$, where x contains at most n occurrences of symbols from N_B it holds $|x|_{a_i} = |y|_{a_i}$ for at least one i. Therefore either $\gamma(x)$ or $\gamma(y)$ is not in L_{n+1} and $L(\Sigma_n, w_n) \ne L_{n+1}$. □

3.2. Hierarchy based on the number of the types of agents

This subsection illuminates the division of the PEG systems according to the number of types of agents in the definition of the system. We will show that the number of the types of agents in the definition of the PEG system (thus the cardinality of the set N_B) affects the generative power of the PEG system. To prove that such a hierarchy exists, we need to find language L which is not $0L$ language and for which holds that L cannot be generated by the PEG system with n types of the agents and L can be generated by the PEG system with $n+1$ types of agent. Not surprisingly, the structure of the language is similar to the language from the previous subsection.

Definition 3.4. A PEG system with at most n types of the agents is such a PEG system $\Sigma = (V_E, N_B, E, B_1, \dots, B_m)$, where $n \le |N_B|$ and is denoted as PEG^n.

The class of languages defined by the PEG^n system is denoted as $\mathcal{L}(\mathrm{PEG}^n)$.

Let us show simple example first.

Example 3.5. Consider the language $L^2 = \{a^i b^i a^j b^j a^k : i \geq 1, j = 2i, k = 3i\}$ and PEG system $\Sigma^2 = (\{a,b\}, \{[1],[2]\}, E, B_1, B_2)$, where:

- $E = (\{a,b\}, \{a \to a, b \to b\})$,
- $B_1 = ([1], \{[1]a \to a[1]ab, [1]b \to aa[1]bbb\})$,
- $B_2 = ([2], \{a[2] \to aaaa[2]\})$.

Let the axiom of the system be $w^2 = [1]abaaa[1]bbbaaa[2]$, then
$[1]abaaa[1]bbbaaa[2] \Rightarrow a[1]abbaaaa[1]bbbbaaaaaa[2] \Rightarrow aa[1]ab^3a^6[1]b^6a^9[2] \Rightarrow$
$\Rightarrow a^3[1]ab^4a^8[1]b^8a^{12}[2] \Rightarrow a^4[1]ab^5a^{10}[1]b^{10}a^{15}[2]\ldots,$
thus $L(\Sigma^2, w^2) = L^2$, Σ has two types of the agents and therefore $L^2 \in \mathcal{L}(\mathrm{PEG}^t)$ for $t \geq 2$.

To prove $L^2 \in \mathcal{L}(\mathrm{PEG}^2)$ we will verify that L^2 does not belong to $\mathcal{L}(\mathrm{PEG}^1)$, i.e. there is no PEG system Σ with $|N_B| = 1$ and axiom w such that $L(\Sigma, w) = L^2$. Assume contrary that L^2 belongs to $\mathcal{L}(\mathrm{PEG}^1)$ and PEG^1 system $\Sigma^0 = (\{a,b\}, \{[1]\}, E, B_1)$ with axiom $w_0 = u_0[1]v_0[1]x_0[1]y_0$ generates L^2. Reader can verify that $E = (\{a,b\}, P)$ in Σ_0 is deterministic and $P = \{a \to a, b \to b\}$. So, all changes in the environment due to the activity of agent.

Let $u[1]v[1]x[1]y$ be the word derived from the axiom in Σ_0 and $uvxy = a^t b^t a^{2t} b^{2t} a^{3t}$ for some $t \geq 1$. According to definition of the PEG system each agent can react with its right or left hand-side symbol, but not with both at once. This restriction allows us consider following cases:

- Let $u = a^{t-1}, v = ab^t a^{2t}, x = b^{2t} a^{3t}, y = \varepsilon$, then Q_1 must contain rules $[1]a \to a[1]ab, [1]b \to aa[1]bbb, a[1] \to aaaa[1]$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+3} b^{t+1} a^{t+2} b^{t+2} a^{t+3}$, which is not from the L^2.

- Let $u = a^{t-1}, v = ab^t a^{2t}, x = b^{2t}, y = a^{3t}$, then Q_1 must contain rules $[1]a \to a[1]ab, [1]b \to aa[1]bbb, [1]a \to [1]aaaa$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+3} b^{t+1} a^{2t+2} b^{2t+2} a^{3t+3}$, which is not from the L^2.

- Let $u = a^{t-1}, v = ab^t a^{2t}, x = b^{2t}, y = a^{3t}$, then Q_1 must contain rules $[1]a \to a[1]ab, [1]b \to aa[1]bbb, b[1] \to b[1]aaa$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+1} b^{t+1} a^{2t+2} b^{2t+2} aba^{3t}$, which is not from the L^2.

In all cases we got the contradiction so the assumption that L^2 belongs to $\mathcal{L}(\mathrm{PEG}^1)$ does not hold and $L^2 \in \mathcal{L}(\mathrm{PEG}^2)$.

Generally speaking, the fact that agent can consider only one-sided context allows us to increment two same substrings of type $a^i b^i$ by only two different number.

Theorem 3.6. The class of PEG languages with respect to the number of types of the agents forms an infinite hierarchy: $\mathcal{L}(\mathrm{PEG}^n) \subset \mathcal{L}(\mathrm{PEG}^{n+1})$ for $n \geq 1$ (i.e. proper subsets).

P r o o f. Obviously $\mathcal{L}(\text{PEG}^n) \subseteq \mathcal{L}(\text{PEG}^{n+1})$ for $n \geq 1$.

Consider language

$$L^{n+1} = \{a^i b^i a^{2i} b^{2i} \dots a^{(2n-1)i} b^{(2n-1)i} a^{2ni} b^{2ni} a^{(2n+1)i} : i \geq 1\}.$$

We will show that $L^{n+1} \in \mathcal{L}(\text{PEG}^{n+1}) - \mathcal{L}(\text{PEG}^n)$.

Hence we will show that we need at least $n+1$ types of agents to generate the language L^{n+1}, n types of agents are not sufficient.

At first we will construct the PEG system Σ^{n+1} with $n+1$ types of agents which generate the language L^{n+1}: $\Sigma^{n+1} = (V_E, \{[1]\}, E, B_1, B_2, \dots, B_n, B_{n+1})$, where

- $V_E = \{a, b\}$,
- $E = (V_E, \{a \to a, b \to b\})$,
- $B_1 = ([1], \{[1]a \to a[1]ab, [1]b \to aa[1]bbb\})$,
- $B_2 = ([2], \{[2]a \to aaa[2]abbb, [2]b \to aaaa[2]bbbbb\})$,

$\vdots$

- $B_n = ([n], \{[n]a \to a^{2n-1}[n]ab^{2n-1}, [n]b \to a^{2n}[n]bb^{2n}\})$,
- $B_{n+1} = ([n+1], \{a[n+1] \to aa^{2n+1}[n+1]\})$.

The axiom of the PEG^{n+1} is

$$w = [1]abaa[1]bbaa[2]abbbaaaa[2]bbbb \dots a^{2n-2}[n]ab^{2n-1}a^{2n}[n]b^{2n}a^{2n+1}[n+1].$$

The first several derivation steps are:

$$[1]abaa[1]bbaa[2]abbbaaaa[2]bbbb \dots a^{2n-2}[n]ab^{2n-1}a^{2n}[n]b^{2n}a^{2n+1}[n+1]$$

$$\Rightarrow a[1]abba^4[1]b^4a^5[2]ab^6a^8[2]b^8 \dots a^{4n-3}[n]ab^{4n-2}a^{4n}[n]b^{4n}a^{4n+2}[n+1]$$

$$\Rightarrow aa[1]abbba^6[1]b^6a^8[2]ab^9a^{12}[2]b^{12} \dots a^{6n-4}[n]ab^{6n-3}a^{6n}[n]b^{6n}a^{6n+3}[n+1] \Rightarrow \dots.$$

Hence the $L(\Sigma^{n+1}, w) = \{a^i b^i a^{2i} b^{2i} \dots a^{(2n-1)i} b^{(2n-1)i} a^{2ni} b^{2ni} a^{(2n+1)i} : i \geq 1\}$.

To prove that $L^{n+1} \in \mathcal{L}(\text{PEG}^{n+1})$ we will verify that $L^{n+1} \notin \mathcal{L}(\text{PEG}^n)$, i. e. there is no PEG system Σ and axiom w with $|N_B| = n$ such that $L(\Sigma, w) = L^{n+1}$.

Assume contrary that L^{n+1} belongs to $\mathcal{L}(\text{PEG}^n)$ and PEG^n system $\Sigma^n = (V_E, N_B, E, B_1, \dots, B_n)$ with axiom w^n generates L^{n+1}. We verify that $E = (V_E, P)$ in Σ^n is deterministic and $P = \{z \to z : z \in \{a, b\}\}$.

It follows from the structure of L^{n+1} that for $z \to \alpha \in P$ we have $\alpha = z^s$ for some s. Assume that P is not deterministic and $z \to z^j \in P$ for $j \geq 0$. Let derivation of $u = a^r b^r a^{2r} b^{2r} \dots a^{(2n-1)r} b^{(2n-1)r} a^{2nr} b^{2nr} a^{(2n+1)r} \in L_{n+1}$ for some r uses the rule $z \to z^j$. Then also the word $v = a^{j(r-2)+1} b^r a^{2(r)} b^{2(r)} \dots a^{r(2n-1)} b^{r(2n-1)}\ a^{2nr} b^{2nr} a^{r(2n+1)} \in L(\Sigma, w)$ but $v \notin L^{n+1}$. Evidently for rules $z \to z^j$ in deterministic P it holds $j = 1$. Otherwise all words from L^{n+1} are not in $L(\Sigma^n, w^n)$.

So, all the changes in the environment are done by the activity of the agents. □

Let $u_0[1]u_1[1]u_2[2]u_3[2]\dots u_{2n-1}[n]u_{2n}[n]u_{2n+2}[o]u_{2n+3}$, where $u_0 = a^{t-1}$, $u_1 = ab^ta^{2t}, u_2 = b^{2t}a^{3t-1}, u_3 = ab^{3t}a^{4t}, \dots,$ $1 \leq o \leq n$, be the word derived from the axiom in Σ^n and $u_0 \dots u_{2n+3} = a^tb^ta^{2t}b^{2t}\dots a^{(2n-1)t}b^{(2n-1)t}a^{2nt}b^{2nt}a^{(2n+1)t}$ for some $t \geq 1$. Without loss of generality we may consider $o = 1$. According to definition of the PEG system each agent can react with its right or left hand-side symbol, but not with both at once. This restriction allows us consider only substring $u_{2n+2}[o]u_{2n+3}$ in following cases:

- Let $u_{2n+2} = b^{2nt-1}, u_{2n+3} = ba^{t(2n+1)}$, then Q_1 must contain rules $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, [1]b \to [1]ba^{2n+1}$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+1}b^{t+1}a^tba^{2n+1}b^t \dots\ a^{2n(t+1)}b^{2n(t+1)}$ $a^{(2n+1)(t+1)}$, which is not from the L^{n+1}.
- Let $u_{2n+2} = b^{2nt}, u_{2n+3} = a^{t(2n+l)}$, then Q_1 must contain rules $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, b[1] \to b[1]a^{2n+1}$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+1}b^{t+1}a^{t+2}b^{t+2}\dots a^{2n(t+1)}b^{2n(t+1)}$ $aaba^{t(2n+1)-1}$, which is not from the L^{n+1}.
- Let $u_{2n+2} = b^{2nt}, u_{2n+3} = a^{t(2n+l)}$, then Q_1 must contain rules $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, [1]a \to [1]aa^{2n+1}$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+2n+1}b^ta^{t+2}b^{t+2}\dots a^{2n(t+1)}b^{2n(t+1)}$ $a^{(2n+1)(t+1)}$, which is not from the L^{n+1}.
- Let $u_{2n+2} = b^{2nt}a^k, u_{2n+3} = a^l$, where $k+l = t(2n+1)$, then Q_1 must contain rules $[1]a \to a[1]ab, [1]b \to aa[1]bbb, [1]a \to [1]aa^{2n+1}$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+2n+1}b^ta^{t+2}b^{t+2}\dots a^{2n(t+1)}$ $b^{2n(t+1)}a^{(2n+1)(t+1)}$, which is not from the L^{n+1}.
- Let $u_{2n+2} = b^{2nt}a^k, u_{2n+3} = a^l$, where $k+l = t(2n+1)$, then Q_1 must contain rules $[1]a \to a[1]ab, [1]b \to aa[1]bbb, a[1] \to a^{2n+1}a[1]$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+2n+1}b^ta^{t+2}b^{t+2}\dots a^{2n(t+1)}$ $b^{2n(t+1)}a^{(2n+1)(t+1)}$, which is not from the L^{n+1}.
- Let $u_{2n+2} = b^{2nt}, u_{2n+3} = a^{t(2n+1)}$, then Q_1 must contain rules $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, a[1] \to a^{2n+1}a[1]$. Such set of rules allows to derive in the next derivation step word w such that $\gamma(w) = a^{t+2n+1}b^{t+1}a^tba^{2n+1}b^t\dots\ a^{2n(t+1)}$ $b^{2n(t+1)}\ a^{(2n+1)(t+1)}$, which is not from the L^{n+1}.

In all cases we got the contradiction so the assumption that L^{n+1} belongs to $\mathcal{L}(\mathrm{PEG}^n)$ does not hold and $L^{n+1} \in \mathcal{L}(\mathrm{PEG}^{n+1})$. Hence $\mathcal{L}(\mathrm{PEG}^n) \subsetneq \mathcal{L}(\mathrm{PEG}^{n+1})$.

4. CONCLUSION

In this paper we have shown that the generative power of the PEG systems depends on the number of types of agents in the definition of the system and on the number of agents present in the environment as well. We have shown the hierarchy of the PEG systems based on both mentioned properties.

ACKNOWLEDGEMENT

Article has been made in connection with project IT4Innovations Centre of Excellence, reg. no. CZ.1.05/1.1.00/02.0070.

Research was also supported by the SGS/24/2013 Project of the Silesian university in Opava.

(Received July 31, 2013)

REFERENCES

[1] E. Csuhaj-Varjú, J. Dassow, J. Kelemen, and Gh. Păun: Grammar Systems – A Grammatical Approach to Distribution and Cooperation. Gordon and Breach, London, 1994.

[2] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, and Gh. Păun: Eco-grammar systems. A grammatical framework for studying lifelike interactions. Artif. Life *3* (1997), 1–28.

[3] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, and Gh. Păun: Eco(grammar) systems – A preview. In: Cybernetics a Systems '94. (R. Trappl, ed.) World Scientific, Singapore 1994, pp. 941–948.

[4] L. Kari, G. Rozenberg, and A. Salomaa: L-systems. In: Handbook of Formal Languages, Vol. 1, (G. Rozenberg and A. Salomaa, eds.) Springer-Verlag, Berlin 1997, pp. 253–324.

[5] A. Kelemenová and M. Langer: Positioned agents in eco-grammar systems. Internat. J. Found. Comput. Sci. *22* (2011), 237–246.

[6] A. Kelemenová and M. Tupý: Monocultures and homogeneous environment in eco-grammar systems. Fund. Inform. *76* (2007), 3, 349–365.

[7] M. Langer: Agents placed in the environment of eco-grammar systems – Positioned eco-grammar systems. In: Pre-Procs. 1st Doctoral Workshop on Mathematical a Engineering Methods in Computer Science (M. Češka et al., eds.) FI MU, Brno 2005, pp. 31–37.

[8] M. Langer and A. Kelemenová: Positioned agents in eco-grammar systems with border markers and pure regulated grammars. Kybernetika *48* (2012), 502–517.

[9] M. Langer and A. Kelemenová: On Positioned eco-grammar systems and pure grammars of Type 0. Neural Network World *2* (2013), 13, 2013, 81–91.

[10] C. Martin-Vide and Gh. Păun: New topics in colonies theory. Grammars *1* (1999), 209–323.

[11] C. Martin-Vide, Gh. Păun: PM-Colonies. Comput. Artificial Intell. *17* (1998), 553–582.

[12] Gh. Păun and A. Salomaa: Families generated by grammars and L systems. In: Handbook of Formal Languages, Vol. 1, (G. Rozenberg and A. Salomaa, eds.) Springer, Berlin, 1997, pp. 811–859.

[13] P. Sosík: On the Hierarchy of Extended Conditional Tabled Eco-Grammar Systems. Grammars *1* (1999), 3, 225–238.

Miroslav Langer, Institute of Computer Science and Research Institute of the IT4 Inovations Centre of Excelence, Silesian University, Bezručovo nám. 13, 746 01 Opava. Czech Republic.

e-mail: miroslav.langer@fpf.slu.cz