Variants of grammar systems: motivations and problems

Alica Kelemenová

Institute of Computer Sciemce, Silesian univerzity in Opava, Opava, Czech Republic,
kelemenova@fpf.slu.cz

Abstract: Grammar systems form an important part of investigation of formal aspects of multi-agent systems. Cooperation, distribution and complexity are basic research topics of grammar systems. Recently the theory covers different types of systems motivated by technology as well as by biology. We present some representative variants of grammar systems, e.g. CD grammar systems, PC grammar colonies and eco-grammar systems. We illustrate different behavior of these systems like sequential, parallel behavior, and their more complex combinations, based on the motivation to introduce systems. We recapitulate some representative research topics, typical and interesting results, including provocative open problems and present rich references.

1 GRAMMAR SYSTEMS

Grammar systems theory is a field of theoretical computer science that studies systems of finite collections of formal grammars generating a formal language. Each grammar works on a string that represents an environment. Grammar systems can thus be used as a formalization of decentralized or distributed systems of agents in artificial intelligence.

Study of grammar systems started in the 90-ties of the last century with motivation of black board architecture. The notion of the cooperating distributed grammar system (CD grammar system, for short) was introduced in [13] and in [16]. First models of grammar systems theory are mainly motivated by distributed AI. The concept of CD grammar systems was proposed as a syntactic model of the blackboard architecture of problem solving, where several independent agents work together on the solution of a problem by cooperating with each other only by modifying the common blackboard representing the current state of problem solving.

In a CD grammar system the agents are generative grammars and the global data base is represented by a string. The agents take turns in rewriting this string according to a given cooperation strategy. The successful problem solving is achieved by generating a terminal word. CD grammar systems demonstrated that complex behavior, i.e. complex languages can be generated by simple grammars using a simple cooperation strategy. For an overview about CD grammar systems see [15] and [30].

While CD grammar systems use sequential rewriting (in each derivation step only one grammar is active), parallel communicating grammar systems (PC grammar systems, for short) introduce parallelism into grammar systems theory.

PC grammar systems are motivated by another problem the classroom model. In this model each agent can operate only on its own „notebook" and only one component, the master can use the blackboard. The agents work in parallel and they can communicate by sending their „notes" to each other. Parallel communicating grammar systems (PCGS, for short) were introduced in Paun Santean in order to investigate concepts like parallelism, synchronization and data communication with formal language theoretic means. In this PC grammar system the components are generative grammars working on their own strings in parallel and communicating with each other by sending their strings by request. The language generated by the system consists of words of the master.

Unfortunately, that language families introduced in this fashion are rather intricate from a formal language point of view, even if one restricts oneself to right-linear grammar components. They have rather weak descriptive capacity.

In [33] a variant of PC grammar systems, called PC grammar systems with terminal transmission, were introduced. Right-linear centralized version of these systems has nice formal language theoretic properties: they are closed under union and gsm mappings (in particular, under intersection with regular sets and under homomorphism) slight variant is also closed under concatenation and star. Their power lies between that of n-parallel grammars introduced by Wood and that of matrix languages of index n, and their relation to equal matrix grammars of degree n is discussed. Membership problem and questions concerning grammatical inference of these systems are studied.

In version of PCGS introduced by Csima, so-called query symbols are cosidered formally as terminal symbols (and not as nonterminal symbols). Right-linear PCGS with terminal transmission have rather nice formal language properties, including simple hierarchical relations to well-known regulated formal language classes and complexity classes.

Recently, there are many variations of GS studied. We mention for example papers [1], [2] and [4]. For further models and inspirations see references at the end of this paper.

While the concept of CD and PC grammar systems is inspired by distributed AI, decentralized AI and Artificial

Life (AL) give motivations for models like colony and eco-grammar system. Decentralized AI deals with the study of multi-agent systems of autonomous agents. In these systems the communication and cooperation is minimized, there is no predefined strategy (unlike in distributed AI), the properties of the system emerge only from the intensive interaction of the agents and the environment. AL studies man-made systems that exhibit behaviors characteric of natural living systems. Its models have similar properties to those of decentralized AI: they consist of a population of simple agents reacting on a common environment without any master component, which would direct their behaviour or coordinate their work. Life-like features are the result only of the interaction of the components and the environment.

2 COLONIES

The colony, introduced in [36] is motivated mainly by the behavior of reactive agents in robotics. This model retains the idea of CD and PC grammar systems of having grammars as components working together but in a colony these grammars are simple regular grammars generating finite languages and there is no cooperation between them. The model has an environment component represented by a string. The state of the system can be changed only by the actions of the agents; the environment is passive, it does not change its state autonomously. Due to the lack of any predefined strategy, each grammar participates in the rewriting whenever it can, conflicts are resolved nondeterministically. The language generated by the colony is the set of all the possible states of the environment. Several derivation modes and acceptance styles were introduced and studied. See [19] and [10] for formal definitions and results. Colonies proved to show emergent behavior indeed: the components are very simple regular grammars, but powerful, large language classes can be generated in this way.

Basic differences among various possibilities of a behavior of a colony, due to the number of components used in one step. The sequential model, parallel models and colonies working in teams are discussed. For further derivation modes for the original colony as well as results for different variants on terminal alphabet we refer to [19], [41], [38].

In next section we will deal with parallel colonies in order to discuss problem concerning equality of language classes for week and strong parallel derivation modes.

2.1 Parallel Colonies

Basic ideas of the colony were formalized the following way:

Definition 1. *A colony* $\mathscr{C}$ *is a 4-tuple* $\mathscr{C} = (V,T,\mathscr{F},\stackrel{x}{\Rightarrow})$, *where*

- V *is an alphabet of the colony,*
- $T \subseteq V$ *is a terminal alphabet of the colony,*
- $\mathscr{F} = \{(S_i,F_i) : S_i \in V, F_i \subseteq (V - S_i)^*,$ F_i is finite, $1 \leq i \leq n$ $\}$.
  A pair (S_i,F_i) *is called i-th component of* $\mathscr{C}$, S_i *is its start symbol and* F_i *is the language of i-th component.*
- $\stackrel{x}{\Rightarrow} \subseteq V^* \times V^*$ *is a derivation step relation.*

A relation $\stackrel{x}{\Rightarrow}$ on strings represents an elementary string transformation realized by components and called a derivation step.

As usual, $\stackrel{x}{\Rightarrow}^*$ stays for a reflexive and transitive closure of $\stackrel{x}{\Rightarrow}$ called derivation. It represents string transformations realized by finite sequences of derivation steps.

Language $L_x(\mathscr{C},w_0)$ determined by a colony $\mathscr{C} = (V,T,\mathscr{F},\stackrel{x}{\Rightarrow})$ and an initial string (axiom) $w_0 \in V^*$ consists of all terminal strings derived from the axiom, i.e. $L_x(\mathscr{C},w_0) = \{v |\ w_0 \stackrel{x}{\Rightarrow}^* v,\ v \in T^*\}$.

By $\mathscr{L}(COL_x)$ we denote the class of all languages generated by colonies with $\stackrel{x}{\Rightarrow}$ derivation.

In parallel colonies introduced in [29] several agents can be active simultaneously according the principle: Components of a colony which can work on the actual string must work simultaneously. 'Component can work' means here that the start symbol of the component is present in the environment. Each active component has to rewrite one occurrence of its start symbol. No competition conflict occurs in derivation in the case when all components have different start symbols. In an opposite case competition conflict occurs when some components have identical start symbols and there are not enough occurrences of that symbol in the environment.

Two ways were proposed to solve activity of components in this case. In strongly competitive case $\stackrel{sp}{\Rightarrow}$ the derivation is blocked for the lack of start symbols in an actual string.

In weakly competitive case $\stackrel{wp}{\Rightarrow}$ the derivation continues and maximal number of components (nondeterministically chosen) is used to rewrite the string. Formally:

Definition 2. *Let* $\mathscr{C} = (V,T,\mathscr{F},\stackrel{sp}{\Rightarrow})$ *be a colony and* $x,y \in V^*$. *We write* $x \stackrel{sp}{\Rightarrow} y$ *if and only if*

- $x = x_1 S_{i_1} x_2 S_{i_2} \dots x_k S_{i_k} x_{k+1}$,
- $y = x_1 z_{i_1} x_2 z_{i_2} \dots x_k z_{i_k} x_{k+1}$ *and* $z_{i_j} \in F_{i_j}, 1 \leq j \leq k$, *where* $i_u \neq i_v$ *for* $u \neq v, 1 \leq u,v \leq k$,
  (one component is allowed to rewrite at most one occurrence of its start symbol),
- $|x|_S > 0$ *implies that for each component* (S,F) *in* $\mathscr{F}$, *there is* $i_j, 1 \leq i_j \leq k$ *such that* $(S,F) = (S_{i_j},F_{i_j})$.

To introduce weak parallelism of colony $\mathscr{C}$ we denote by $m(S)$ the number of components in $\mathscr{C}$ with start symbol S. Maximal possible number of components will be active in one weak parallel derivation step.

Definition 3. *Let $\mathscr{C} = (V, T, F, \overset{wp}{\Rightarrow})$ be a colony and $x, y \in V^*$. We write $x \overset{wp}{\Rightarrow} y$ if and only if*

- $x = x_1 S_{i_1} x_2 S_{i_2} \dots x_k S_{i_k} x_{k+1}$,
- $y = x_1 z_{i_1} x_2 z_{i_2} \dots x_k z_{i_k} x_{k+1}$ *and* $z_{i_j} \in F_{i_j}, 1 \le i_1, \dots i_k \le n, i_u \ne i_v$ *for all* $u \ne v, 1 \le u, v \le k$,

  (one component rewrites at most one occurrence of its start symbol)
- $|x|_S = r \ge 0$ *implies that for* $l = min\{r, m(S)\}$ *different components* $(S, F_{i_j}), 1 \le j \le l$ *from* $\mathscr{F}$ *chosen nondeterministically, we have* $(S, F_{i_j}) = (S_{i_j}, F_{i_j})$.

Colonies with strongly competitive parallel derivation as well as colonies with weakly competitive parallel derivation can produce some context sensitive languages. Following [15], [29] we have

Proposition $\mathscr{L}(COL_{sp}) \subseteq \mathscr{L}(COL_{wp})$,

$$\mathscr{L}(COL_{wp}) \subseteq \mathscr{L}(ET0L)_{[1]} \subset \mathscr{L}(M, ac) \text{ and}$$

$$\mathscr{L}(COL_{sp}) \subseteq \mathscr{L}(M, ac),$$

where $\mathscr{L}(ET0L)_{[1]}$ are one limited ET0L languages and $\mathscr{L}(M, ac)$ is the class of matrix languages with appearance checking.

An important **open problem** from 1998 is the equality of the language classes $\mathscr{L}(COL_{sp})$ and $\mathscr{L}(COL_{wp})$.

To illustrate the topic we present parallel colonies for some non context-free languages in next examples.

Example 1. sp mode

(i) $\{ww : w \in \{0,1\}^+\} \in \mathscr{L}(COL_{sp})$.

Let $\mathscr{C} = (\{A, B, C, D, X, 0, 1\}, \{0, 1\}, \mathscr{F}, \overset{sp}{\Rightarrow})$, where
$\mathscr{F} = \{(A, \{0B, 1C\}), (A, \{0C, 1B\}), (B, \{A, X\}), (C, \{A, X\}), (X, \{\varepsilon\}), (X, \{\varepsilon\})$.

There are two components with start symbol A and two components with start symbol X *in* $\mathscr{F}$. Derivation in $\mathscr{C}$ starting with AA is blocked for a string with only one A or only one X. For example

$AA \overset{sp}{\Rightarrow} 0B1B \overset{sp}{\Rightarrow} 0A1B$

An example of derivation of terminal word is

$AA \overset{sp}{\Rightarrow} 0B0C \overset{sp}{\Rightarrow} 0A0A \overset{sp}{\Rightarrow} 01C01B \overset{sp}{\Rightarrow} 01X01X \overset{sp}{\Rightarrow} 0101$.

Colony $\mathscr{C}$ with axiom AA generates the language

$L_{sp}(\mathscr{C}, AA) = \{ww : w \in \{0,1\}^+\}$.

Another colony $\bar{\mathscr{C}}_{sp}$ produces the same language

$\bar{C}_{sp} = (\{A, B, C, D, X, 0, 1\}, \{0, 1\}, \mathscr{F}, \overset{sp}{\Rightarrow}))$, *where*
$\mathscr{F} = \{(A, \{0B, 1C\}), (A, \{0B, 1C\}), (B, \{A, X\}), (B, \{A, X\}), (C, \{A, X\}), (C, \{A, X\}), (X, \{\varepsilon\}), (X, \{\varepsilon\})\}$.

Derivation gives for example

$AA \overset{sp}{\Rightarrow} 0B0B \overset{sp}{\Rightarrow} 0A0A \overset{sp}{\Rightarrow} 01C01C \overset{sp}{\Rightarrow} 01X01X \overset{sp}{\Rightarrow} 0101$

Let the derivation start with $AA \overset{sp}{\Rightarrow} 0B1C$.

Colony has two components for B and just one B in the actual string so derivation is blocked. Derivation ends with terminal words only if two identical "nonterminals" are in all the sequential forms.

$L(\bar{\mathscr{C}}_{sp}, AA) = \{ww : w \in \{0,1\}^+\}$.

(ii) $\{a^i b^i c^i \mid i \ge 0\} \in \mathscr{L}(COL_{sp})$.

Consider colony

$\mathscr{C}_{sp} = (\{S, A, B, C, D, E, F, a, b, c\}, \{a, b, c\}, \mathscr{F}, \overset{sp}{\Rightarrow})$, where
$\mathscr{F} = \{(A, \{aD, X\}), (B, \{bE, X\}), (C, \{cF, X\}), (D, \{A\}), (E, \{B\}), (F, \{C\}), (X, \{\varepsilon\}), (X, \{\varepsilon\}), (X, \{\varepsilon\})\}$.

Only possibility to derive terminal word is to use components rewriting X. Three components with identical start symbol X guarantee that terminal string is derived using these components simultaneously. Derived strings have prescribed structure and terminal string is derived in synchronous rewriting two occurrences of X in the first example and three occurrences of X in the second example. Otherwise less occurrences of X in derived string stops derivation unsuccessfully. So $L(\mathscr{C}_{sp}, S) = \{a^i b^i c^i \mid i \ge 0\}$.

Example 2. wp mode

(i) $\{ww : w \in \{0,1\}^+\} \in \mathscr{L}(COL_{wp})$.

Let $\mathscr{C} = (\{P, Q, R, X, Y, B, 0, 1\}, \{0, 1\}, \mathscr{F}, \overset{wp}{\Rightarrow})$, where
$\mathscr{F} = \{ (P, \{0QX, 1RX, Y\}), (P, \{0RY, 1QY, X\}), (Q, \{P\}), (Q, \{B\}), (R, \{\varepsilon\}), (R, \{B\}), (X, \{\varepsilon\}), (X, \{B\}), (Y, \{\varepsilon\}), (Y, \{B\})\}$.

Strings with at most one occurrence of Q, R, X, Y can be rewritten to terminal words. Pairs of these symbols produce non active B and block the derivation.

Successful derivation:

$PP \overset{wp}{\Rightarrow} 0QX0RY \overset{wp}{\Rightarrow} 0P0P \overset{wp}{\Rightarrow} 01RX01QY \overset{wp}{\Rightarrow} 01P01P \overset{wp}{\Rightarrow} 01X01Y \overset{wp}{\Rightarrow} 0101$.

Blocked derivation:

$PP \overset{wp}{\Rightarrow} 0QX1QY$ or $PP \overset{wp}{\Rightarrow} 0QXX$.

Language determined by a colony is

$L_{wp}(\mathscr{C}, PP) = \{ww : w \in \{0,1\}^+\}$.

(ii) $\{\{a^i b^i c^i \mid i \ge 0\} \in \mathscr{L}(COL_{wp})$.

Consider colony

$\mathscr{C} = (\{A, B, C, D, E, F, X, Y, Z, H, a, b, c\}, \{a, b, c\}, \mathscr{F}, \overset{wp}{\Rightarrow})$, where
$\mathscr{F} = (A, \{aDXX, aYZ\}), (B, \{bEYY, bXZ\}), (C, \{cFZZ, cXY\}), (D, \{A\}), (E, \{B\}), (F, \{C\}), (X, \{\varepsilon\}), (X, \{\varepsilon\}), (X, \{H\}), (Y, \{\varepsilon\}), (Y, \{\varepsilon\}), (Y, \{H\}), (Z, \{\varepsilon\}), (Z, \{\varepsilon\}), (Z, \{H\})$.

Successful derivation:

$ABC \overset{wp}{\Rightarrow} aDXXbEYYcFZZ \overset{wp}{\Rightarrow} aAbBcC \overset{wp}{\Rightarrow} aaYZbbXZcXY \overset{wp}{\Rightarrow} aabbcc$

Blocked derivation

$ABC \overset{wp}{\Rightarrow} aDXXbEYYcXY$

So $L_{wp}(\mathscr{C}, S) = \{a^i b^i c^i \mid i \ge 0\}$.

(iii) $\{a^i b^j c^k \mid i, j, k \ge 0, i \ne j, \; j \ne k, \; i \ne k\} \in \mathscr{L}(COL_{wp})$.

Consider colony

$\mathscr{C}_{wp} = (\{S, A, B, C, D, E, F, a, b, c\}, \{a, b, c\}, \mathscr{F}, \overset{wp}{\Rightarrow})$, where
$\mathscr{F} = \{(A, \{aD, X\}), (B, \{bE, X\}), (C, \{cF, X\}), (D, \{A\}), (E, \{B\}), (F, \{C\}), (X, \{\varepsilon\}), (X, \{Y\}), (X, \{Y\})\}$

A successful derivation in the colony ends by rewriting all occurrences of X by ε. There are at most three occurrences of X in sentential forms produced by the colony but only words containing at most one X can be rewritten to terminal words.
Successful derivation:
$ABC \overset{wp}{\Rightarrow} XbEcF \overset{wp}{\Rightarrow} bBcC \overset{wp}{\Rightarrow} bXccF \overset{wp}{\Rightarrow} bccC \overset{wp}{\Rightarrow} bccX \overset{wp}{\Rightarrow} bcc$
Blocked derivation
$ABC \overset{wp}{\Rightarrow} aDbEcF \overset{wp}{\Rightarrow} aAbBcC \overset{wp}{\Rightarrow} aXbXcX \overset{wp}{\Rightarrow} aYbYc$

$L(\mathscr{C}_{wp}, S) = \{a^i b^j c^k \mid i, j, k \geq 0, i \neq j \text{ and } j \neq k \text{ and } i \neq k\}.$

2.2 PM-colonies

The PM-colonies, as collection of agents located on string environment with ability to change their neighboring agents or environment, were introduced and studied in [47], [48]. Introduction of a PM-colony is based on the following assumptions:

- the environment is described by a string of symbols; names of agents are symbols appearing in this string (agents are parts of the environment with specific places); agents can modify symbols of the environment description (hence also the symbols identifying other agents, which means that the agents can act on each other);

- agents are only able to perform point mutations (similar to those ones appearing in the genetic area): erase one symbol, insert one symbol, substitute one symbol for another one;

- these actions take place only in the strict vicinity of the symbol representing the agent; in order to allow mobility, one also considers move actions by which the agent symbols can be interchanged with a neighboring symbol (irrespective whether or not this one is a name of another agent);

- all actions described above (point mutations and moves) depend on the smallest nontrivial neighborhood of the agent: the environment symbols adjacent to the left and to the right (the boundary markers, at the ends of the environment); actions take place simultaneously for all agents present in the environment (this pair of neighboring symbols form the context of the agent); conflicts are solved by a priority relation among agents: when two agents have a common context or, even more, one agent is placed in the context of another one, then the agent with higher priority will act; in the case of equal priority a deadlock appears;

- agents remain unchanged in all the elementary actions described above (point mutations and moves), except the following possibilities: an agent can remove its own name ("death"), can introduce and remove the name of another agent ("birth" and "death"; thus, an agent cannot change its own name or the name of another agent).

Two variants how to define and solve conflicts in models were discussed together with their influences to the generative power of the PM colonies.

1) PM-colony, original model
Let agents A_i and A_j appear in the string. We say that A_i and A_j are in conflict if they have common context or one agent is a part of context of the other agent. Let A_i,A_j and A_k be agents. If A_i and A_j are in conflict and A_j and A_k are in conflict at the same time, then we say that agents A_i and A_k are in conflict too. We say that appearance of agent A_i in context aA_ib in the environment w is active, if and only if there exists some rule with left side of form (a, Ai, b) and it holds that A_i is not in conflict with any other agent, or it is in conflict with one or more agents but it has the greatest priority. Generative power of PM colonies introduced above was studied in [20] with following results:

PM-colonies are able to generate some context sensitive languages, ($\mathscr{L}(COL_{PM})$ - $\mathscr{L}(CF) \neq \emptyset$) but on the other side there are finite languages that cannot be generated by them ($\mathscr{L}(FIN) - \mathscr{L}(COL_{PM}) \neq \emptyset$). This way of introduction conflicts and activity of agents seems to be very restrictive. It can occur that agents in the common environment do not affect each other, but they are in the conflict caused by another agent(s). And if there is no agent with the greatest priority in chain of such agents, all agents in the string become inactive.

2) PM-colony with significant context
Another possibility to introduce the conflict of agents in PM colony was investigated in [5]. One can consider "significant context" of the agent. Significant context is a substring of length five, containing two letters before and two letters after the agent. It will determine conflict between agents and activity of agents as follows: The agent is not in conflics in the case if there is no other agent in its significant context, or if it has the greatest priority with respect to the all other agents in its significant context. AGent is active, if it is not in conflict and the is a rule to rewrite it with respect to his actual left and righ neighbour.In all other cases the agent is no active. Such attempt to the activity of agents differs from that one introduced above. For example, consider the string $AaBaA$ where A has greater priority then B. Both occurrences of agent A are active by considering significant context, while according to the original definition from previous section all agents are inactive. To distinguish new derivation approach we will use PMs instead of PM. For the generative power od PMs colonies we $RE = \mathscr{L}(COL_{PMs})$ and $\mathscr{L}(COL_{PMs}) - \mathscr{L}(COL_{PM}) \neq \emptyset$.

3 ECO-GRAMMAR SYSTEMS

The model of an eco-grammar system was introduced in [17] and presented in detail in [18]. It realizes an attempt to create formal grammar specification for investigation of the interplay between the environment and agents in systems like ecosystems using framework of grammar systems [15]. Eco-grammar systems can be used to model some aspects of the behavior of cooperating communities of agents operating in a common dynamic environment.

The model, an eco-grammar system, consists of the interconnected parts of *environment* and *agents or compo-*

nents. The environment is described by a string, developing in totally parallel manner according to the derivation mode of an 0L system (i.e.using interactionless rules). Each agent is described by a string developing according to the derivation mode of an 0L system as well. Moreover, using specific action rules the agent can locally change the environment, and the actual state of the environment can influence the development of agents and the states of agents can influence the development of the environment by choice of active action rules.

Basic information on eco-grammar systems can be found in overview papers [39], [38], [7]. Variants of EG systems motivated by PM colonies were studied in [42, 45, 44]. Eco-colonies, the variant of EG systems motivated by colonies were studied by Š. Vavrečková. Power of eco-colonies were compared with that of colonies and EG systems in [60],[62] and [63].

One of the videly investigated topic of EG systems is their team behaviour. The concept of a team, introduced into grammar systems theory in [35], appears in eco-grammar systems theory in two different forms: prescribed teams discussed in [3] and team derivation modes [20]. In the case of prescribed teams the system contains the specification of those groups of agents which can work together in a derivation step. Derivation mode k prescribes that in each derivation step exactly k agents work. Thus, in this derivation mode any k agents can form a team and can work together. If we consider simple eco-grammar systems i.e. systems, where the agents, independently of the actual state, can execute all possible actions on the environment we obtain different results for systems with 0L behavior and E0L behavior of the environment.

The number of the agents in the system and the number of the agents working in a derivation step form hierarchy of language classes in non-extended case. In an extended simple eco-grammar system these hierarchies collapse.

In next part we take attention to special cases of eco-grammar systems with identical components called monocultures.

Monocultures

Eco-grammar systems with identical agents are called the monocultures. Several results concerning the generative power and the hierarchy and/or incomparability of these systems according to the number of components were presented in [43], [40], [7] and [9]. Relation between EG languages and languages of monocultures seems to be surprising.

Typical results concerning monocultures are following:

1. Monocultures over unary alphabet are as powerful as eco-grammar systems over unary alphabet.
This is a consequence of the fact that every unary language of an eco-grammar system can be generated by an eco-grammar systems with one agent and therefore by a monoculture.

2. Finite languages as well as semiunary languages are languages of monocultures, where a semiunary language is a language over at least binary alphabet and each word of the language is the string of identical letters.

3. Eco-grammar systems with passive environment can be simulated by monocultures.

4. An equivalent weak monoculture can be constructd to each eco-grammar system, where a weak monoculture is an EG systems with identical agents, each of which can start in different states.

This gives rise the question what happends if we are looking for the monocultures where components are started with the same axiom.

(In)equality of the generative power of eco-grammar systems and monocultures is still an open problem. Note, that we put no restriction to the number of components of monoculture which simulate the behaviour of EG system. Number of components in these systems can be different.

Perhaps the most interesting partial result related to it is that monocultures do not restrict substantially the generative power of the eco-grammar systems in the following sense:

5. One can add a single word to each language of an eco-grammar system to obtain a language of a monoculture.

This word is the axiom of the corresponding monoculture. In [40] we are looking for suitable candidates for axiom inside the language generated by an eco-grammar system. If we could find such a word for each at least binary EG languages then equality holds.

Acknowledgement

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602.

References

[1] Arthi, K., Krithivasan, K., Csuhaj-Varjú, E.: On Rule-Number Complexity of Components of Probabilistic Cooperating Distributed Grammar Systems. Journal of Automata, Languages and Combinatorics 7(4): 433-446 (2002)

[2] Autebert, J.-M.: Some results about centralized PC grammar system. Theo. Comput. Sci. **215** (1999) 383—398

[3] ter Beek, M.H., Eco-grammar systems and colonies. In: Grammatical Models of Multi-Agent Systems, 1999

[4] ter Beek, M.H., Csuhaj-Varjú, E., Holzer, M., Vaszil, G.: Co-operating Distributed Grammar Systems: Components with Nonincreasing Competence. Computation, Cooperation, and Life. LNCS **6610** Springer (2011)

[5] Blažek, M.: Řešení konfliktů v PM-koloniích. Diplomová práce. Ústav informatiky, Slezská univerzita, Opava 2008

[6] Bordihn, H., Dassow, J., Vaszil, G.: Grammar systems as language analyzers and recursively enumerable languages. In: Proc. FCT'99, LNCS **1684** (1999) 136–147.

[7] Csima, J.: On Extended Simple Eco-grammar Systems. Acta Cybern. (1998) 359–373

[8] Csima, J.: On Hybrid Eco-rewriting Systems. Grammars **1**(1999) 239-253

[9] Csima, J.: Two remarks on variants of simple eco-grammar systems. Acta Cybern. **14** (2000) 569-582

[10] Csuhaj-Varjú, E.: Colonies: a multi-agent approach to language generation. Extended finite state models of language. Cambridge University Press (1999) 208-225

[11] Csuhaj-Varjú, E.: On size complexity of context-free returning parallel communicating grammar systems. Where Mathematics, Computer Science, Linguistics and Biology Meet (2001 37-49

[12] Csuhaj-Varjú, E.: CD Grammar Systems: Competence and Confidence. Computation, Cooperation, and Life 2011: 57-69

[13] Csuhaj-Varju, E., Dassow, J.: On cooperating-distributed grammar systems, J. Inform. Process. Cybernet. 26 (1990) 49-63

[14] Csuhaj-Varjú, E., Dassow, J.: On the Size of Components of Probabilistic Cooperating Distributed Grammar systems. Theory Is Forever 2004: 49-59

[15] Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Păun, G.: Grammar Systems. A Grammatical Approach to Distribution and Cooperation. Gordon & Beach, London, 1994.

[16] E. Csuhaj-Varju and J. Kelemen, Cooperating grammar systems: a syntactical framework for blackboard model of problem solving. In: Proc. AIICSR 1989 (1989) 121 –127.

[17] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenovâ., G. Pâun: Eco(grammar) systems. A preview. In: Proc. 12th European Meeting on Cybernetics and System Research, Vienna, 1994, R. Trappl, ed., World Scientific, Singapore, 1994, 941–948

[18] Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A., Păun, G.: Eco-grammar Systems. Grammatical Framework for Studying Lifelike Interactions. Artificial Life **3** (1997) 1–28

[19] Csuhaj-Varjú, E., Kelemenová, A.: Languages of Colonies. *T*heoretical Computer Science 134 (1994) 119-130

[20] Csuhaj-Varjú, E., Kelemenová, A.: Team behaviour in eco-grammar systems. Theoretical Computer Science **209** (1998) 213–224

[21] Csuhaj-Varjú, E., Vaszil, Gy.: On the Number of Components and Clusters of Non-returning Parallel Communicating Grammar Systems. DCFS 2011: 121-134

[22] Csuhaj-Varjú, E., Vaszil, Gy.: On the Descriptional Complexity of Context-Free Non-returning PC Grammar Systems. Journal of Automata, Languages and Combinatorics 15(1/2): 91-105 (2010)

[23] Csuhaj-Varjú, E., Vaszil, Gy.: On the Size Complexity of Non-Returning Context-Free PC Grammar Systems. DCFS 2009: 91-100

[24] Csuhaj-Varjú, E., Vaszil, Gy.: Parallel communicating grammar systems with bounded resources. Theor. Comput. Sci.276(1-2): 205-219 (2002)

[25] Csuhaj-Varjú, E., Paun, G.: Limiting the team size in cooperating grammar systems. Bulletin of the EATCS 49: 198-201(1993)

[26] Csuhaj-Varju, E., Vaszil, Gy.: On context-free parallel communicating grammar systems: Synchronization, communication, and normal forms. Theoret. Comput. Sci. **255** (2001) 511–538

[27] Csuhaj-Varju, E., Vaszil, G.: Parallel communicating grammar systems with incomplete information communication. Develop. Language Theory (2001), 381–392

[28] Dassow, J.: On cooperating distributed grammar systems with competence based start and stop conditions. Fund. Inform. **76** (2007) 293–304

[29] Dassow, J., Kelemen, J., Păun, G.: On Parallelism in Colonies. Cybernetics and Systems 24, 37–49 (1993)

[30] Dassow, J., Păun, G., Rozenberg, G.: Grammar systems. In: Handbook of Formal Languages, Springer, Berlin (1997) 155–214

[31] Dumitrescu, S., Păun, G.: On the power of parallel communicating grammar sys-tems with right-linear components. RAIRO Informatique th´eorique et Applications/Theoretical Informatics and Applications **31** (1997) 331–354

[32] Fernau, F: PC grammar systems with terminal transmission. In: Proc.International Workshopon Grammar Systems, Silesian University at Opava, (2000) 229–252

[33] Fernau, H.: Parallel communicating grammar systems with terminal transmission. Acta Inform. **37** (2001) 511–540

[34] Fernau, H., Holzer, M.: Graph-controlled cooperating distributed grammar systems with singleton components. In: Proc. Third Internat. Workshop on Descriptional Complexity of Automata, Grammars, and Related Structures, Vienna (2001) 79-90

[35] Kari, L., Mateescu, A., Paun, G., Salomaa, A: Teams in cooperating grammar systems. J. Exp. Theor. Artif. Intell. 7(4): 347-359 (1995)

[36] Kelemen, J.,Kelemenová, A.: A Grammar-theoretic Treatment of Multiagent Systems. Cybernetics and Systems **23** (1992) 621–633

[37] Kelemen, j., Kelemenová, A., Martín-Vide, C., Mitrana, Colonies with limited activation of components, Theoretical computer science 244 (2000) 289–298.

[38] Kelemenová, A., Kelemen, J.: From colonies to eco(grammar) systems. An overview. In: *Proc. Important Results and Trends in Theoretical Computer Science, Graz, 1994.*, Lecture Notes in Computer Science 812, Springer Verlag, Berlin, 1994, 211–231

[39] Kelemenová, A.: Eco-grammar systems. In: *Formal languages and applications*, Studies in Fuzziness and Soft Computing 148. Springer, Berlin, 2004, 311–322

[40] Kelemenová, A., Bartík, T.: Monocultures in eco-grammar systems. AFL 2008: 208-219

[41] Kelemenová, A.,Csuhaj-Varjú, E.: Languages of Colonies. Theoretical Computer Science **134** (1994) 119–130

[42] Kelemenová, A., Langer, M.: Positioned Agents in Eco-Grammar System. IJFCS 22, (2011) 237-–246

[43] Kelemenová,A., Tupý, M.: Monocultures and homogeneous environment in eco-grammar systems. *Fundamenta Informaticae 76* (2007) 349–365

[44] Langer, M.: On hierarchy of the positioned eco-grammar systems Kybernetika 50 (5), 696-705 2 2014

[45] Langer, M., Kelemenová, A.: Positioned agents in eco-grammar systems International Journal of Foundations of

Computer Science **4**(2011) 237-246

[46] Lukáš, R., Meduna, A.: Multigenerative grammar systems and matrix grammars. Kybernetika **46** (2010) 68-82

[47] Martin-Vide, C., Paŭn, G.: PM-Colonies, Computers and Artificial Intelligence **17** (1998)553–582.

[48] Martin-Vide, C., Paŭn, G.: New topics in colonies theory, Grammars **1** (1999) 209–323.

[49] Mateescu, A.: Teams in cooperating distributed grammar systems; an overview, Proc. Decentralized Intelligent and Multi-Agent Systems, DIMAS'95, Cracow, (1995) 309–323

[50] Meduna, A.: Two-way metalinear PC grammar systems and their descriptional complexity. Acta Cybernet **16** (2003) 126—137

[51] Meduna, A., Lukas, R.: Multigenerative grammar systems. Schedae Inform. **15** (2006) 175—188

[52] Paun, Gh.: On the synchronization in parallel communicating grammar systems. ActaInf **30** (1993)351–367

[53] Paun, Gh., Santean, L.: Parallel communicating grammar systems: the regular case. Ann.Univ. Bucharest, Ser. Matem.-Inform. **38** (1989) 55–63

[54] Păun,G., Salomaa, A., Vicolov, S.: On the generative capacity of parallel communicating grammar systems. Internat. J. Comput. Math. **45** (1992) 45–59

[55] Rozenberg, G., Salomaa, A. eds.: Handbook of Formal Languages. Springer, Berlin 1997

[56] Rovan, B., Slast'an, M.: Eliminating Communication by Parallel Rewriting. Developments in Language Theory (2001) 369-278

[57] Sebestyén, P., Sosík, P: Modelling Multiple Robots in Space: An Adaptive Eco-Grammar System. Fundamenta Informaticae, 2007

[58] Subramanian,K. G., Venkat, I., Csuhaj-Varjú, E.: On the power of permitting features in cooperating context-free array grammar systems. Discrete Applied Mathematics 161(15): 2328-2335 (2013)

[59] Vaszil, G.: On simulating non-returning PC grammar systems with returning systems. Theoret. Comput. Sci. **209** (1998)

[60] Vavrečková, Š.: Eko-kolonie. In: Kognice a umělý život V, Silesian University, Opava, (2005) 601–612

[61] Vavrečková, Š.: Eco-colonies. In:MEMICS 2006, Proc. of the 2nd Doctoral Workshop, University of Technology, FIT, Brno, (2006) 253–259

[62] Vavrečková, Š.: Properties of Eco-colonies. In:Information Systems and Formal Models 2007, A. Kelemenová, Silesian University, Opava, (2007) 235–242

[63] Vavrečková, Š, Kelemenová, A.: Properties of Eco-Colonies. In: Proc. Automata for Cellular and Molecular Computing, G. Vaszil (ed.), MTA SZTAKI, Budapest (2007) 129–143