International Journal of Foundations of Computer Science

DYNAMICALLY FORMED CLUSTERS OF AGENTS IN ECO-GRAMMAR SYSTEMS

KATALIN ANNA LÁZÁR

Department of Information Systems, Eötvös Loránd University
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
lazarka@inf.elte.hu

ERZSÉBET CSUHAJ–VARJÚ

Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13–17, 1111 Budapest, Hungary
and
Department of Algorithms and Applications, Eötvös Loránd University
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
csuhaj@sztaki.hu

ANDRÁS LŐRINCZ

Department of Information Systems, Eötvös Loránd University
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
lorincz@inf.elte.hu

GYÖRGY VASZIL

Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13–17, 1111 Budapest, Hungary
vaszil@sztaki.hu

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

In this paper we extend the conditions of dynamic team constitution in simple eco–grammar systems, motivated by the bottom–up–clustering algorithm. The relationships of simple eco–grammar systems formed according to the newly introduced conditions to each other as well as to certain language classes of the Chomsky hierarchy and L systems are established. We prove that any recursively enumerable language can be obtained as the intersection of a regular language and the language of simple eco–grammar systems where the active teams are organized according to different conditions of team constitution. We also propose some further research directions.

Keywords: bottom–up–clustering algorithm, simple eco–grammar systems, recursively enumerable languages, Lindenmayer systems

1. Introduction

In this article we extend the conditions of dynamic team constitution [13] in simple eco–grammar systems [11]. We are inspired by idea that the behavior of the agents participating in the bottom–up–clustering algorithm [2, 7, 16, 22] bears some similarities to that of the components of these variants of eco–grammar systems.

Communities of interacting and communicating agents create dynamically evolving network topologies. The study of the properties of complex networks is the goal of current research [25]. In particular, a plethora of networks exhibit a small average distance between vertices, a phenomenon called small–world effect, typical of random graphs [5], coupled with local clustering properties, characteristic of ordered lattices [33]. Moreover, many random networks are scale–free [1], in the sense that they exhibit a power–law distribution of the degree.

In effect, the bottom–up–clustering algorithm is based on the small–world concept and the idea that complex networks have high clustering coefficients. In [23] a unified view is proposed to describe both global and local traits of networks by means of a single measure, the connectivity length. It has a precise meaning in terms of information propagation, since the dissemination of information occurs in a highly efficient manner in networks characterized by small global and local connectivity length.

Teams correspond to clusters in the formal language theoretic construction introduced in this paper. We emphasize that our model is purely a syntactical one employed to describe how the bottom–up–clustering algorithm works. We examine the boundaries of the model at a pure syntactic level. Apparently, more sophisticated traits could be added, which might shed a new light on some further aspects of the algorithm.

2. Related Works

An eco–grammar system aims at modelling the interplay between the environment and the agents in complex systems such as ecosystems [11]. It intends to capture some aspects of multi-agent systems [10]: the grammars correspond to very simple autonomous agents and the generated language to the behavior of the system. Herein we consider a restricted variant of eco–grammar systems, called simple eco–grammar systems [11]. Briefly, a simple eco–grammar system consists of some agents and an environment. The agents are represented by a set of context–free rules, the environment by a set of developmental rules. At any moment of time, the behavior of the system is described by the state of the environment. The environmental state is altered by derivation steps. At each derivation step the agents act on the environment through the application of one of their context-free rules in parallel with the environment that replaces the remaining symbols via its developmental rules. The evolution of the system can be characterized by the sequences of strings obtainable through a sequence of derivation steps from the initial string representing the environment.

The idea of considering teams of grammars in grammar systems has been employed in a series of papers. Teams can be either predefined [3, 4, 12, 28, 32] or formed dynamically [13, 31]. Our choice of simple eco–grammar systems with dynamically formed teams of agents is motivated by the fact that the entities are very simple autonomous agents with limited capabilities, i.e. memoryless and reactive entities working in a dynamically changing problem space. The transformation of the problem space corresponds to the alternation of the environment. The modification of the environment is the result of the joint actions of the agents and the development of the environment. Through their actions the agents contribute to the solution of the problem. The development of the environment expresses the change of the level of difficulty of the problem. The agents are organized into teams: those that are able to disseminate information are allowed to participate in the creation of a given team.

In this paper the relationships of simple eco–grammar systems formed according to the newly introduced conditions to each other as well as to certain language classes of the Chomsky hierarchy and L systems are established. We prove that any recursively enumerable language can be obtained as the intersection of a regular language and the language of simple eco–grammar systems where the active teams are organized according to different conditions of team constitution.

3. Formal Language Prerequisites

The reader is assumed to be familiar with the basics of formal language theory, for further details consult [19, 29, 30]. For the sake of legibility, only the most important notions used throughout this article are revised.

For an alphabet V, we denote by V^* the set of words over V, by V^+ the set of all nonempty words, i.e. $V^+ = V^* \setminus \{\lambda\}$, where λ is the empty string. Let $(x)_U$ be the string obtained through the erase of the symbols that are not in U. Let $length(x)$ denote the length of x and $alph(x)$ the set of symbols occurring in $x \in V^*$. Moreover, for $L \subseteq V^*$, let $alph(L) = \bigcup_{x \in L} alph(x)$. For a finite set A, $card(A)$ stands for the number of elements of A. The set of natural numbers is denoted by $\mathcal{N}$ and $\mathcal{N}_0 = \mathcal{N} \cup \{0\}$.

A *Chomsky grammar* is a quadruple denoted by $G = (N, T, S, P)$, where N is the nonterminal alphabet, T is the terminal alphabet, $N \cap T = \emptyset$, $S \in N$ is the start symbol and P is a (finite) set of rewriting rules. We say that $x \in (N \cup T)^*$ directly derives $y \in (N \cup T)^*$, written as $x \Longrightarrow y$, iff $x = x_1 u x_2, y = x_1 v x_2$ for some $u \to v \in P$. The reflexive and transitive closure of $\Longrightarrow$ is denoted by $\Longrightarrow^*$ and the language generated by G is defined by $L(G) = \{x \in T^* \mid S \Longrightarrow^* x\}$.

Throughout the paper we will employ generative mechanisms different from Chomsky grammars.

By a *random context grammar*, we mean a quadruple $G = (N, T, S, P)$, where N is the nonterminal alphabet, T is the terminal alphabet, $N \cap T = \emptyset$, $S \in N$ is the start symbol and P is a finite set of rewriting rules of the form $(Q, R) : A \to \alpha$, where

$A \in N, \alpha \in (N \cup T)^*$ and Q, R are subsets of N. For two strings $x, y \in (N \cup T)^*$, y is said to be directly derived from x through the application of $(Q, R) : A \to \alpha \in P$, denoted by $x \Longrightarrow_P y$, if $x = x_1 A x_2, y = x_1 \alpha x_2$, $x_1, x_2 \in (N \cup T)^*$, and each element of Q and no element of R appears in $x_1 x_2$. Q is called the set of permitting symbols, whereas R the set of forbidding symbols associated to $A \to \alpha$ in production $(Q, R) : A \to \alpha \in P$. If Q or R contains only one element, then we may write B instead of $\{B\}$ in the production. For any random context grammar G, an equivalent random context grammar G' can be constructed with $card(Q) \leq 1$ and $card(R) \leq 1$.

A *pure context-free grammar* is a pair $\gamma = (V, \omega, P)$, where V is an alphabet, $\omega \in V^+$ is the axiom of the grammar and P is a finite set of rewriting rules of the form $a \to v$, where $a \in V$, $v \in V^*$. For two strings x and y in V^*, we say that x yields y in a direct derivation step by using a rule in P, written as $x \Rightarrow_P y$, if $x = x_1 a x_2$, $y = x_1 v x_2$, where $x_1, x_2 \in V^*$ and $a \to v \in P$. P is *complete*, if for each $a \in V$ there exists a rule $a \to x$ in P.

A 0L system (an *interactionless Lindenmayer system*) is a triplet $G = (V, \omega, P)$, where V, ω and P are defined in the same way as in the case of pure context-free grammars and P is supplemented with the completeness condition. The rules of P are called the developmental rules of G. For $z_1, z_2 \in V^*$, we write $z_1 \Longrightarrow z_2$ (if necessary to specify P, we write $\Longrightarrow_P$), if $z_1 = a_1 a_2 \ldots a_r, z_2 = x_1 x_2 \ldots x_r$, for $a_i \to x_i$ in P, $1 \leq i \leq r$.

A T0L system (a *tabled interactionless Lindenmayer system*) is an $(n+2)$-tuple $G = (V, \omega, P_1, \ldots, P_n)$, $n \geq 1$, where $G_i = (V, \omega, P_i)$ is a 0L system for each $i, 1 \leq i \leq n$. For $z_1, z_2 \in V^*$, we write $z_1 \Longrightarrow z_2$, if $z_1 \Longrightarrow_{P_i} z_2$ for some $i, 1 \leq i \leq n$.

4. Basic Definitions

In the sequel, we describe the bottom–up–clustering algorithm in terms of simple eco–grammar systems with dynamically formed teams of agents. For this reason, we extend the conditions of team constitution introduced in [13].

A simple eco–grammar system with n agents, $n \geq 1$, is a construct $\Gamma = (V, P_E, R_1, \ldots, R_n, \omega_E)$, where

- V is a finite alphabet, the alphabet of the system,
- P_E is a finite and complete set of pure context-free rules over V, the set of developmental rules of the environment,
- $R_i, 1 \leq i \leq n$, is a finite set of pure context-free rules over V, the set of action rules of the i-th agent,
- $\omega_E \in V^+$ is the initial state of the environment.

A string over V^* is a state of the environment or more briefly an environmental state. A simple eco–grammar system functions through the change of its environmental states. The environmental states are altered both by the action rules of the agents and by the developmental rules of the environment.

The set of symbols appearing on the left-hand side of the rules of R_i is denoted by $dom(R_i)$, i.e. $dom(R_i) = \{a \mid a \to x \in R_i\}$. In fact, $dom(R_i)$ corresponds to the set of symbols that can be modified by an action of the $i-$th agent.

By a team in a simple eco–grammar system Γ we mean a set of agents.

A simple eco–grammar system works in such a manner that each member of the given team substitutes exactly one occurrence of a symbol with a word by means of its action rules in the current environmental state, whereas the other symbols are rewritten by the developmental rules of the environment. We note that two different derivation modes can be defined based on the two versions of parallel rewriting for colonies introduced in [14]. If the strong derivation mode is used then each agent of the given team has to apply one of its productions. It signifies that should an agent be unable to rewrite a symbol of the current sentential form, then the underlying team cannot be employed. In the weak derivation mode, however, only those agents that are members of the same team and able to rewrite a symbol of the current sentential form, have to do so. Herein, we will use only the strong variant, which we will not indicate separately.

Formally, for a simple eco-grammar system $\Gamma = (V, P_E, R_1, \dots, R_n, \omega_E)$, a team $\mathcal{T} = \{R_{i_1}, \dots, R_{i_s}\}$, $i_j \in \{1, \dots, n\}$, $1 \leq j \leq s \leq n$, and two environmental states ω, ω', we define the (strong) direct derivation step (written by $\omega \models_{\mathcal{T}} \omega'$) as follows:

- $\omega = x_0 a_1 x_1 \dots a_s x_s$ and $\omega' = y_0 z_1 y_1 \dots z_s y_s$, $a_h \in V_E, x_j, y_j, z_h \in V_E^*$, $1 \leq h \leq s, 0 \leq j \leq s$,
- $a_h \to z_h \in R_{i_h}$ for all $1 \leq h \leq s$, where if $h \neq h'$ for some $1 \leq h, h' \leq s$, then $i_h \neq i_{h'}$ and
- $y_j = x_j$ is either the empty word, or $x_j \Rightarrow_{P_E} y_j$, $0 \leq j \leq s$, i.e. x_j directly derives y_j in the 0L manner.

So as to introduce the extension of the different dynamic team constitution modes proposed in [13], we review the concept of the level of competence/excitation of an agent.

Definition 1. *Let $\Gamma = (V, P_E, R_1, \dots, R_n, \omega_E)$ be a simple eco–grammar system as above. For an environmental state $\omega \in V^*$, the level of competence/excitation of R_i, $1 \leq i \leq n$, with respect to ω is defined as follows: $lev(R_i, \omega) = card(alph(\omega) \cap dom(R_i))$, i.e. the number of different symbols from ω belonging to $dom(R_i)$. We say that R_i is competent with respect to ω, if $lev(R_i, \omega) \geq 1$ holds.*

Informally, the level of competence/excitation of an agent with respect to the environmental state expresses the number of different symbols occurring in the environmental state that can be replaced by that agent.

Definition 2. *Let $\Gamma = (V, P_E, R_1, \dots, R_n, \omega_E)$ be a simple eco–grammar system as above, $\omega \in V^+$, and $\mathcal{T} = \{R_{i_1}, R_{i_2}, \dots, R_{i_m}\}$, with $i_j \in \{1, 2, \dots, n\}$, $1 \leq j \leq m \leq n$, be a team of agents in Γ, where each member of $\mathcal{T}$ is competent with respect to ω. We say that $\mathcal{T}$ is formed according to condition $d^{\Diamond q}$ with respect to ω, $q \in \mathcal{N}_0$, $\Diamond \in \{\leq, =, \geq\}$, if for all $R_{i_j}, R_{i_k} \in \mathcal{T}$, $i_j, i_k \in \{1, 2, \dots, n\}$, $1 \leq j, k \leq m \leq n$,*

$|lev(R_{i_j},\omega) - lev(R_{i_k},\omega)|\diamondsuit q$ and there is no $R_l, 1 \leq l \leq n$, such that R_l is not an element of $\mathcal{T}$, R_l is competent with respect to ω and for all member R_{i_r} of $\mathcal{T}$, $i_r \in \{1,2,\ldots,n\}$, $1 \leq r \leq m \leq n$, $|lev(R_{i_r},\omega) - lev(R_l,\omega)|\diamondsuit q$ holds.

In Definition 2 those agents that are competent with respect to the environmental state and differ from each other in their level of competence/excitation by at most/exactly/at least q (cases $\leq, =$ and $\geq$) belong to the same team. Observe that singleton teams, i.e. teams consisting of one member, may also be formed and not necessarily one team can satisfy the condition of team constitution in team mode $d^{\diamondsuit q}$, $\diamondsuit \in \{\leq, =, \geq\}$, $q \in \mathcal{N}_0$. Furthermore, in team constitution mode $d^{=q}$, $q \in \mathcal{N}$, the teams can only have two members.

Definition 3. *Let $\Gamma = (V, P_E, R_1, \ldots, R_n, \omega_E)$ be a simple eco–grammar system as above, $\omega \in V^+$, and $\mathcal{T} = \{R_{i_1}, R_{i_2}, \ldots, R_{i_m}\}$ a team of agents in Γ, where $i_j \in \{1,2,\ldots,n\}$, $1 \leq j \leq m \leq n$, and each R_{i_j} is competent with respect to ω. Then $\mathcal{T}$ is formed according to condition $c^{\diamondsuit q}$ with respect to ω, where $\diamondsuit \in \{\leq, =, \geq\}$, $q \in \mathcal{N}_0$, if $card(dom(R_{i_j})) - lev(R_{i_j},\omega)\diamondsuit q$, $1 \leq j \leq m \leq n$, and there is no $R_l, 1 \leq l \leq n$, such that R_l is not an element of $\mathcal{T}$, R_l is competent with respect to ω and $card(dom(R_l)) - lev(R_l,\omega)\diamondsuit q$.*

Definition 3 could be interpreted as follows: an agent is a member of a given team provided that the agent is competent with respect to the environmental state and the cardinality of the set of symbols appearing on the left-hand side of the rules of the agent differs from its level of competence/excitation by at most/exactly/at least q (cases $\leq, =$ and $\geq$).

Definition 4. *Let $\Gamma = (V, P_E, R_1, \ldots, R_n, \omega_E)$ be a simple eco–grammar system as above, $\omega \in V^+$, $\mathcal{T} = \{R_{i_1}, R_{i_2}, \ldots, R_{i_m}\}$, $i_j \in \{1,2,\ldots,n\}$, $1 \leq j \leq m \leq n$, a team of agents in Γ, where each member of $\mathcal{T}$ is competent with respect to ω. Let $\emptyset \neq V_B, V_C \subseteq V$, $V_B \triangle V_C$, where $\triangle \in \{\subseteq, =, \supseteq\}$. Then $\mathcal{T}$ is formed with respect to ω according to the condition $t^{\triangle V_B}$, $\triangle \in \{\subseteq, =, \supseteq\}$, if for all $R_{i_j} \in \mathcal{T}$, $i_j \in \{1,2,\ldots,n\}$, $1 \leq j \leq m \leq n$, $(\omega)_{V_C} \in dom(R_{i_j})^+$ and there is no $R_l, 1 \leq l \leq n$, such that R_l is not an element of $\mathcal{T}$, R_l is competent with respect to ω and $(\omega)_{V_C} \in dom(R_l)^+$ is satisfied.*

In Definition 4 in case of team constitution mode $t^{=V_B}$, $\emptyset \neq V_B \subseteq V$, the agent is considered to be the member of the team, if the agent is competent with respect to the environmental string and the string obtained from the environmental string through the deletion of the letters not belonging to a certain subset V_B is an element of the set of strings that can be produced using the set of symbols appearing on the left-hand side of the rules of the given agent. Team constitution modes $t^{\subseteq V_B}$ and $t^{\supseteq V_B}$, where $\emptyset \neq V_B \subseteq V$, may be interpreted analogously.

Condition $d^{=0}$ in Definition 2 is the same as condition e, condition $c^{=0}$ in Definition 3 as condition c and condition $t^{=V_B}$, $\emptyset \neq V_B = V$, in Definition 4 as condition t in [13].

Definitions 2, 3 and 4 describe the different cases of cluster formation. A plethora of measures have been proposed in the literature on which cluster creation is based [2, 7, 8, 16, 20, 26].

Definition 5. *For a simple eco–grammar system* $\Gamma = (V, P_E, R_1, \ldots, R_n, \omega_E)$*, as above, and for two environmental states* ω, ω'*, we say that* ω *directly derives* ω' *in* Γ *in team derivation mode* α*, where* $\alpha \in \{d^{\Diamond q}, c^{\Diamond q}, t^{\triangle V_B} \mid \Diamond \in \{\leq, =, \geq\}, \triangle \in \{\subseteq, =, \supseteq\}, q \in \mathcal{N}_0, \emptyset \neq V_B \subseteq V\}$*, denoted by* $\omega \stackrel{\alpha}{\Longrightarrow}_\Gamma \omega'$*, if one of the following holds:*

- *either* $\omega \models_\mathcal{T} \omega'$ *for some team* $\mathcal{T}$ *formed according to condition* α *in* Γ*,*
- *or, if such a team does not exist, then* $\omega \Longrightarrow_{P_E} \omega'$*, i.e.* ω *directly derives* ω' *in the 0L manner.*

The reflexive and transitive closure of relation $\stackrel{\alpha}{\Longrightarrow}_\Gamma$ *is denoted by* $\stackrel{\alpha}{\Longrightarrow}^*_\Gamma$*. If no confusion arises, then* Γ *can be omitted from the notation.*

The language of a simple eco–grammar system is the set of all environmental states that are reachable from the initial configuration by a sequence of direct derivation steps.

Definition 6. *The language generated by a simple eco–grammar system* $\Gamma = (V, P_E, R_1, \ldots, R_n, \omega_E)$*,* $n \geq 1$*, in team derivation mode* α*, for* $\alpha \in \{d^{\Diamond q}, c^{\Diamond q}, t^{\triangle V_B} \mid \Diamond \in \{\leq, =, \geq\}, \triangle \in \{\subseteq, =, \supseteq\}, q \in \mathcal{N}_0, \emptyset \neq V_B \subseteq V\}$*, is defined by* $L(\Gamma, \alpha) = \{y \mid \omega \stackrel{\alpha}{\Longrightarrow}^*_\Gamma y\}$*.*

We illustrate how simple eco–grammar systems work in various team modes through an example.

Example 7. *Let* $\Gamma = (\{a_1, a_2, \ldots a_n\}, P_E, R_1, \ldots, R_n, a_1 a_2 \ldots a_n)$*, where*

$$P_E = \{a_1 \to a_1, a_2 \to a_2, \ldots, a_n \to a_n\}, \text{ and}$$
$$R_i = \{a_i \to a_i a_i\}, \text{ for } 1 \leq i \leq n.$$

Notice that if $f_1 \in \{d^{=0}, d^{\leq 0}, d^{\geq 0}, d^{\leq q_1}, c^{=0}, c^{\leq 0}, c^{\geq 0}, c^{\leq q_1} \mid q_1 \geq 1\}$*, then* $L(\Gamma, f_1) = \{a_1^m a_2^m \ldots a_n^m \mid m \geq 1\}$*, which is not a context–free language provided that* $n \geq 3$*. It can be seen that* $card(dom(R_i)) = 1$*,* $1 \leq i \leq n$*. Initially, the level of competence of an agent is* $lev(R_i, a_1 a_2 \ldots a_n) = 1$*, which does not alter during the derivation. It can be verified also that* $L(\Gamma, f_2) = \{a_1 a_2 \ldots a_n\}$ *and* $L(\Gamma, f_3) = \{a_1 \ldots a_i^m \ldots a_n \mid m \geq 1\}$*, where* $f_2 \in \{d^{=q_1}, d^{\geq q_1}, c^{=q_1}, c^{\geq q_1}, t^{\supseteq \{a_i\}}, t^{\triangle \{a_{j_1}, a_{j_2}, \ldots a_{j_k}\}} \mid q_1 \geq 1, \triangle \in \{\subseteq, =, \supseteq\}, \{j_1, j_2, \ldots j_k\} \subseteq \{1, 2, \ldots n\}, 2 \leq k \leq n, j_q \neq j_p, p \neq q, 1 \leq p, q \leq k\}$*, and* $f_3 \in \{t^{=\{a_i\}}, t^{\subseteq \{a_i\}} \mid 1 \leq i \leq n\}$*. Note that both* $L(\Gamma, f_2)$ *and* $L(\Gamma, f_3)$ *are regular languages, moreover,* $L(\Gamma, f_2)$ *is finite.*

Let $\Gamma = (V, P_E, R_1, \ldots, R_n, \omega_E)$, $n \geq 1$, be a simple eco–grammar system as above. In the sequel, the class of languages generated by simple eco–grammar systems with at most n agents using the α team mode derivation is denoted by $EGL(n, \alpha)$, where $\alpha \in \{d^{\Diamond q}, c^{\Diamond q} \mid \Diamond \in \{\leq, =, \geq\}, q \in \mathcal{N}_0\}$.

For an alphabet V and for some $\emptyset \neq V_B \subseteq V$, $\triangle \in \{\subseteq, =, \supseteq\}$, we denote by $EGL(n, t^{\triangle V_B})$ the class of languages produced by simple eco–grammar systems with at most n agents employing the $t^{\triangle V_B}$ team mode derivation.

By definition we consider a 0L system as a simple eco–grammar system with no agent.

Furthermore, we set $EGL(\alpha) = \bigcup_{n \geq 0} EGL(n, \alpha)$, $\alpha \in \{d^{\Diamond q}, c^{\Diamond q} \mid q \in \mathcal{N}_0, \Diamond \in \{\leq, =, \geq\}\}$ and $EGL(t^{\triangle V_B}) = \bigcup_{n \geq 0} EGL(n, t^{\triangle V_B})$, for some $\emptyset \neq V_B \subseteq V$, $\triangle \in \{\subseteq, =, \supseteq\}$.

5. Hierarchies and Relationships

In this section we investigate the language hierarchies induced by the number of agents, the computational power of simple eco–grammar systems working in the team derivation modes above in comparison with other generative mechanisms such as certain Chomsky grammars and L systems.

5.1. *Hierarchies*

Our aim is to establish whether or not the language hierarchies induced by the number of agents are infinite.

Theorem 8. *Language hierarchies* $EGL(n-1, c^{\Diamond q}) \subseteq EGL(n, c^{\Diamond q})$*,* $\Diamond \in \{\leq, =, \geq\}$*,* $q \geq 0$*, and* $EGL(n-1, \alpha) \subseteq EGL(n, \alpha)$*, where* $n \geq 2$*,* $\alpha \in \{d^{=0}, d^{\geq 0}, d^{\leq q}, d^{\geq 1} \mid q \geq 0\}$*, are infinite.*

Proof. To prove the statement for the team modes $d^{=0}, d^{\geq 0}, d^{\leq q_1}$, $c^{=0}, c^{\geq 0}$ and $c^{\leq q_1}$, $q_1 \geq 0$, consider the language $L = \{a_1^k \dots a_n^k \mid k \in \mathcal{N}\}$, which can be generated in all of the team modes above by the simple eco–grammar system defined in Example 7.

L can also be produced in team mode $d^{\geq 1}$ by a simple eco–grammar system similar to the simple eco–grammar system presented in Example 7 except for the definition of agent R_i, which should be modified as follows: $R_i = \{a_j \to a_j^2 \mid 1 \leq j \leq i\}$, for all $2 \leq i \leq n$.

In order to support our claim, observe that $lev(R_i, \alpha) = i$ for each sentential form α, which signifies that the team consisting of all agents has to be employed in each derivation step. The only possible way of guaranteeing that all agents will be able to work is to choose rule $a_i \to a_i^2$ for agent R_i, $1 \leq i \leq n$.

To generate L in the team modes $c^{=q}$, $c^{\geq q}$, $q \geq 1$, consider the system

$$\Gamma = (\{a_1, \dots, a_n, Y_1, \dots, Y_q\}, P_E, R_1, \dots, R_n, a_1 \dots a_n),$$

where $P_E = \{a_i \to a_i \mid 1 \leq i \leq n\} \cup \{Y_j \to Y_j \mid 1 \leq j \leq q\}$ and $R_i = \{a_i \to a_i^2\} \cup \{Y_j \to Y_j \mid 1 \leq j \leq q\}$, for all $1 \leq i \leq n$.

Assume now that L can be produced in any of the above mentioned team modes by a system with less than n agents, for $n \geq 2$. Since the number of each letter has to be increased by one in each derivation step, either some of the agents have to augment the number of more than one letters, or the number of at least one letter has to be multiplied by the environment. Strings that are not in L can be obtained in both cases, which leads to a contradiction. The detailed proof is left to the reader. $\square$

5.2. *Finite Languages*

We prove that finite languages satisfying certain conditions can be generated by simple eco–grammar systems functioning in some of the above team modes.

Theorem 9. *Let V be an alphabet. Then, for any finite language $L = \{x_1, \ldots, x_n\}$, where $x_i \in V^*$, $1 \leq i \leq n$,*

- *in team derivation mode $d^{=q}$, $q \in \mathcal{N}$, it holds that $L \in EGL(d^{=q})$, provided that $q + 1 \leq card(alph(x_i))$ for all i, $1 \leq i \leq n$;*
- *in team derivation modes $d^{\Diamond 0}$ and $d^{\leq q}$, $\Diamond \in \{\leq, =, \geq\}$, $q \in \mathcal{N}$, it can be verified that $L \in EGL(d^{\Diamond 0})$ and $L \in EGL(d^{\leq q})$;*
- *in team derivation modes $c^{=q}$ and $c^{\leq q}$, $q \in \mathcal{N}_0$, it can be proved that $L \in EGL(c^{=q})$ and $L \in EGL(c^{\leq q})$;*
- *there is V_B, $\emptyset \neq V_B \subseteq V$, in the $t^{\triangle V_B}$, $\triangle \in \{\subseteq, =, \supseteq\}$, team derivation mode, such that the statement $L \in EGL(t^{\triangle V_B})$ is valid.*

Proof. In the first place, it is easy to see that $L(\Gamma, t^{\triangle V_B}) = L(\Gamma, d^{\Diamond 0}) = L(\Gamma, d^{\leq q}) = L$, $\Diamond \in \{\leq, =, \geq\}$, $q \in \mathcal{N}$, $\emptyset \neq V_B = V$, where $\Gamma = (V, \{a \to \lambda \mid a \in V\}, \{a \to x_i \mid a \in V, 1 \leq i \leq n\}, x_1)$.

Secondly, it can be shown that $L(\Gamma, d^{=q}) = L$, $q \in \mathcal{N}$, where $q + 1 \leq card(alph(x_i))$ for all i, $1 \leq i \leq n$. Let us assume that $alph(L) = \{a_1, \ldots, a_{k+q}\}$, $k, q \in \mathcal{N}$. Indeed, the simple eco–grammar system $\Gamma = (V, \{a \to \lambda \mid a \in V\}, (R_i)_{i=1}^{k+q}, (R_{\{j_1,\ldots,j_{q+1}\}})_{j=1}^{M}, x_1)$, where $M = \binom{k+q}{q+1}$, $R_i = \{a_i \to \lambda\}$, $1 \leq i \leq k+q$, $R_{\{j_1,\ldots,j_{q+1}\}} = \{a_{j_1} \to x_s, \ldots, a_{j_{q+1}} \to x_s \mid x_s \in L, 1 \leq s \leq n, \{j_1, \ldots, j_{q+1}\} \subseteq \{1, \ldots, k+q\}, j_l \neq j_{l'}, \text{ if } l \neq l'\}$, and $j_z \neq j'_z, 1 \leq z \leq q+1$, if $j \neq j'$, produces L in team derivation mode $d^{=q}$, $q \in \mathcal{N}$.

Finally, we will demonstrate that $L \in EGL(c^{=q})$ and $L \in EGL(c^{\leq q})$, $q \in \mathcal{N}$. (Case $c^{=0}$ is proven in [13], a similar argument is valid for team modes $c^{\leq 0}$ and $c^{\geq 0}$). Suppose that $alph(L) = \{a_1, \ldots, a_r\}, r \geq 1$, and symbols $Y_1, \ldots, Y_q$ are not in $alph(L)$. In fact, the simple eco–grammar system $\Gamma = (V, \{a \to \lambda, Y_j \to \lambda \mid a \in V, 1 \leq j \leq q\}, (R_i)_{i=1}^{r}, x_1)$, where $R_i = \{a_j \to x_s, Y_1 \to x_s, \ldots, Y_q \to x_s \mid 1 \leq j \leq i, 1 \leq s \leq n\}, 1 \leq i \leq r$, generates L in team derivation modes $c^{=q}$ and $c^{\leq q}$, $q \in \mathcal{N}$. $\square$

5.3. *Regular and Context–Free Languages*

Herein we show that the families of languages generated by simple eco–grammar systems functioning in the extended team modes, defined in Section 2, are incomparable with the family of regular languages and context–free languages.

Theorem 10. *The following assertions can be verified:*

- *for each* $\vartheta \in \{d^{\Diamond q}, c^{\Diamond q} \mid \Diamond \in \{\leq, =, \geq\}, q \in \mathcal{N}_0\}$*, the family* $EGL(\vartheta)$ *is incomparable with the family of regular languages and context-free languages;*
- *for each alphabet* V *with* $card(V) \geq 2$*, there exist* V_B*,* $\emptyset \neq V_B \subseteq V$*, such that the family* $EGL(t^{\triangle V_B})$*,* $\triangle \in \{\subseteq, =, \supseteq\}$*, is incomparable with the family of regular languages and context-free languages.*

Proof. We first note that $L = \{a^{2^n} \mid n \geq 0\}$ is in $EGL(\alpha)$ for any $\alpha \in \{c^{\Diamond 0}, c^{\leq q}, d^{\Diamond 0}, d^{\leq q} \mid \Diamond \in \{\leq, =, \geq\}, q \geq 1\}$, but this language is not context-free. Indeed, the simple eco–grammar system $\Gamma = (\{a\}, \{a \to a^2\}, \{a \to a^2\}, a)$ generates L in any team mode α as above.

Secondly, we may observe that L is also in $EGL(\beta)$ for any $\beta \in \{c^{=q}, c^{\geq q} \mid q \geq 1\}$. In fact, it can be produced by the simple eco–grammar system

$$\Gamma = (\{a, b_1, \ldots, b_q\}, P_E, R_1, a),$$

where $P_E = \{a \to a^2, b_i \to b_i \mid 1 \leq i \leq q\}$, $R_1 = \{a \to a^2, b_i \to b_i \mid 1 \leq i \leq q\}$ in any team mode β as above.

Thirdly, we may conclude that $L = \{a^{2^n} b_1 \ldots b_q \mid n \geq 0\}$ is in $EGL(\gamma)$ for any $\gamma \in \{d^{=q}, d^{\geq q} \mid q \geq 1\}$, though this language is not context-free. In effect, the simple eco–grammar system

$$\Gamma = (\{a, b_1, \ldots, b_q\}, P_E, R_1, R_2, ab_1 \ldots b_q),$$

where $P_E = \{a \to a^2, b_i \to b_i \mid 1 \leq i \leq q\}$, $R_1 = \{a \to a^2\}$, and $R_2 = \{a \to a^2, b_i \to b_i \mid 1 \leq i \leq q\}$ generates L in any team mode γ as above.

Finally, notice that $L = \{a_1^{2^n} a_2^{2^n} \ldots a_k^{2^n} \mid n \geq 0\}$ is in $EGL(\delta)$ for any $\delta = t^{\triangle V_B}$, $\triangle \in \{\subseteq, =, \supseteq\}$, $\emptyset \neq V_B \subseteq V$, since it can be produced by the simple eco–grammar system

$$\Gamma = (\{a_1, \ldots, a_k\}, \{a_i \to a_i^2 \mid 1 \leq i \leq k\}, \{a_i \to a_i^2 \mid 1 \leq i \leq k\}, a_1 a_2 \ldots a_k)$$

in any of team mode $t^{\triangle V_B}$, $\triangle \in \{\subseteq, =, \supseteq\}$, for $V_B = \{a_1, \ldots, a_k\}$.

On the other hand, the language $\{ab^n, ba^n \mid n \geq 1\}$ cannot be generated by any simple eco–grammar system in any of the studied team modes ϑ, $\vartheta \in \{d^{\Diamond q}, c^{\Diamond q}, t^{\triangle V_B} \mid \Diamond \in \{\leq, =, \geq\}, \triangle \in \{\subseteq, =, \supseteq\}, q \in \mathcal{N}_0, \emptyset \neq V_B \subseteq V\}$. To see this, note that team formation is based on the letter occurrence of the strings, therefore simple eco–grammar systems of the types above cannot produce strings of forms ab^i and ba^i, $i \geq 1$, without generating words of different forms. The detailed proof is left to the reader. □

5.4. *L Systems*

It follows directly from the definitions that the family of 0L languages is included in any of the families $EGL(\alpha)$, where $\alpha \in \{c^{\Diamond q}, d^{\Diamond q}, t^{\triangle V_B} \mid \Diamond \in \{\leq, =, \geq\}, \triangle \in$

$\{\subseteq, =, \supseteq\}, \emptyset \neq V_B \subseteq V, q \in \mathcal{N}_0\}$. According to Example 7 the inclusion is strict for $\alpha \in \{d^{=0}, d^{\geq 0}, d^{\leq q}, c^{=0}, c^{\geq 0}, c^{\leq q} \mid q \in \mathcal{N}_0\}$.

Theorem 11. *The claims below are valid:*

- *for each* $\alpha \in \{d^{\diamond q}, c^{\diamond q} \mid \diamond \in \{\leq, =, \geq\}, q \in \mathcal{N}_0\}$*, the family* $EGL(\alpha)$ *and the family of T0L languages are incomparable;*
- *for each alphabet* V*, there exist* V_B*,* $\emptyset \neq V_B \subseteq V$*, such that the family* $EGL(t^{\triangle V_B})$*,* $\triangle \in \{\subseteq, =, \supseteq\}$*, and the family of T0L languages are incomparable.*

Proof. The proof for the team derivation modes $c^{=0}, d^{=0}$ and $t^{=V_B}$ for $\emptyset \neq V_B = V$ can be found in [13]. An analogous argument is valid for the team modes $d^{\geq 0}, d^{\leq q}$, $c^{\geq 0}, c^{\leq q}$, $t^{\triangle\{a\}}$, $q \in \mathcal{N}_0$, $\triangle \in \{\subseteq, =, \supseteq\}$, and the non-T0L language $L_1 = \{a, a^3\}$, which can be generated by the simple eco–grammar system

$$\Gamma_1 = (\{a\}, \{a \to \lambda\}, \{a \to a^3\}, a).$$

A similar system with alphabet $V = \{a, b_1, \ldots, b_q\}$ and agent $\{a \to a^3, b_i \to b_i \mid 1 \leq i \leq q\}$ produces L_1 in team modes $c^{=q}$ and $c^{\geq q}$.

The non-T0L language $L_2 = \{a_1 a_2 \ldots a_{2+q}, a_1^3 a_2^3 \ldots a_{2+q}^3\}$ can be generated in team modes $d^{=q}$ and $d^{\geq q}$ by the system

$$\Gamma_2 = (\{a_1, \ldots, a_{q+2}\}, \{a_i \to \lambda \mid 1 \leq i \leq q+2\}, R_1, R_2, a_1 \ldots a_{q+2}),$$

where $R_1 = \{a_1 \to a_1^3\}$ and $R_2 = \{a_i \to a_2^3 \cdots a_{q+2}^3 \mid 2 \leq i \leq q+2\}$.

The verification of the fact that the T0L language $L_3 = \{a^{2^n 3^m} \mid n, m \geq 1\} \notin EGL(\alpha)$ for any $\alpha \in \{c^{=0}, d^{=0}, t^{=\{a\}}\}$, can be found in [13]. Analogous considerations may be applied to $\alpha \in \{c^{\leq q}, c^{\geq q}, c^{=q}, d^{\leq q}, t^{\supseteq\{a\}}, t^{\subseteq\{a\}} \mid q \in \mathcal{N}_0\}$. Furthermore, it is clear that L_3 cannot be generated in team modes $d^{=q}$ and $d^{\geq q}$, $q \in \mathcal{N}$, since $alph(L_3) = \{a\}$, whereas the alphabet of a simple eco–grammar system working in these team modes has to contain at least $q+1$ elements. □

6. The Power of Team Cooperation

In the sequel, we will demonstrate that team cooperation leads to quite a large computational power.

In this paper, we will not deal with team modes $c^{=0}, d^{=0}, t^{=V}$, $\emptyset \neq V$, since the verification of the theorems below for these cases can be found in [13]. The simple eco–grammar systems employed in [13] can also be used to support the claim for team modes $c^{\leq 0}, c^{\geq 0}, d^{\leq 0}, d^{\geq 0}, t^{\subseteq V}, t^{\supseteq V}$, $\emptyset \neq V$.

First, let us consider team derivation modes $c^{=q}$ and $c^{\leq q}$, $q \in \mathcal{N}$.

Theorem 12. *A language* L *over an alphabet* T *is recursively enumerable if and only if it can be obtained as* $L = L' \cap T^*$ *for some* $L' \in EGL(\alpha)$*, where* $\alpha \in \{c^{=q}, c^{\leq q} \mid q \in \mathcal{N}\}$.

Proof.

Actually, we will show that every language generated by a context-free random context grammar can be expressed in the form claimed by the theorem. Since any recursively enumerable language can be produced by a context-free random context grammar [15, 19], the assertion follows.

Without loss of generality, it may be assumed that L is generated by a context–free random context grammar $G = (N, T, S, P)$, whose productions are of the form $(B, C) : A \to x$, where $A \to x$ is a context–free production and B, C are two non-terminals. Suppose that the productions in P are labelled in a one–to–one manner by numbers from 1 to n and let $N = \{A_1, \ldots, A_r\}$, $r \geq 1$.

We will construct a simple eco–grammar system working in the $c^{=q}$ team mode, $q \in \mathcal{N}$, which simulates the derivation in G.

Let $X, X_i, X'_i, X''_i, Y_j, X_D, X'_D$ and F be new distinct symbols not in $N \cup T$, where $1 \leq i \leq n,\ 1 \leq j \leq q,\ D, D' \in N$. We define

$$\Gamma = (V, P_E, R_0, (R_i, R'_i, R''_i)_{1 \leq i \leq n}, (R_D)_{D \in N}, (R'_D)_{D' \in N}, XS),$$

where

$$V = N \cup T \cup \{X, X_i, X'_i, X''_i, Y_j, F \mid 1 \leq i \leq n,\ 1 \leq j \leq q\} \cup \{X_D, X'_D \mid D, D' \in N\},$$
$$P_E = \{a \to a \mid a \in V \setminus \{\{X''_i \mid 1 \leq i \leq n\} \cup \{X_D, X'_D \mid D, D' \in N\}\}\} \cup$$
$$\{X_D \to F \mid D \in N\} \cup \{X'_D \to F \mid D' \in N\} \cup \{X''_i \to X \mid 1 \leq i \leq n\},$$
$$R_0 = \{X \to X_i X_C, X \to \lambda, Y_j \to F \mid 1 \leq i \leq n,\ 1 \leq j \leq q\},$$

for each $D \in N$

$$R_D = \{X_D \to \lambda, D \to D, Y_j \to F \mid 1 \leq j \leq q - 1\},$$

for each $D' \in N$

$$R'_D = \{X'_D \to \chi, D' \to \psi, Y_j \to F \mid 1 \leq j \leq q,\ \text{where if } D' \neq C \text{ then } \chi = \lambda, \psi = D',\ \text{or}$$
$$\text{if } D' = C \text{ then } \chi = \psi = F\},$$

and for each rule $(B, C) : A \to x$, labelled by i, $1 \leq i \leq n$, we introduce agents

$$R_i = \{X_i \to X'_i, Y_j \to F \mid 1 \leq j \leq q\},$$
$$R'_i = \{X'_i \to X''_i X_{A_1} \ldots X_{A_r}, \delta \to F, Y_j \to F \mid 1 \leq j \leq q,\ \text{where if } B \neq \emptyset \text{ then } \delta = B,\ \text{or}$$
$$\text{if } B = \emptyset \text{ then } \delta = A\},$$
$$R''_i = \{X''_i \to F, A \to x, Y_j \to F \mid 1 \leq j \leq q\}.$$

We explain how a derivation in G can be simulated by some derivation in Γ the $c^{=q}$ team mode, $q \in \mathcal{N}$. Let us assume that at some moment the current state of Γ is $X\omega$, where ω is a sentential form in G. Note that the initial state of Γ is a string of this form. A derivation step in G, where the rule $(B, C) : A \to x$, labelled by i is to be applied may be simulated as follows:

(1) During the first stage, only agent R_0 and agents R_D can work, where $D \in N$ and D appears in ω. It indicates the rule that has to be simulated by rewriting X to $X_i X_C$, $1 \leq i \leq n$, $C \in N$, where C is the forbidding context condition. The work of agents R_D does not change the sentential form.
(2) In the second phase, the simulation is continued by a team formed from agent R_i, agents R_D and agents $R_{D'}$, where $D \in N$, $D \neq C$ occurs in the current sentential form and either $D = C$, D is absent from or $D' \in N$, $D' = C$ is present in the sentential form. R_i rewrites X_i to X'_i and X_C is either erased by agent R_D, if $D \in N$, $D = C$ does not appear in the sentential form or it is substituted for the trap symbol F as a result of the action of agent $R_{D'}$, if $D' \in N$, $D' = C$ is present in the sentential form. The trap symbol F cannot be removed from the string. No other agent can be active during this derivation phase.
(3) At the third stage, a team formed from agent R'_i and agents R_D, where $D \in N$ and D appears in the sentential form is active. R'_i rewrites X'_i to $X''_i X_{A_1} \ldots X_{A_r}$, if there is a symbol B (or symbol A if $B = \emptyset$) in the sentential form. The appearance of the symbol X''_i in the sentential form indicates that the conditions of application of the production labelled by i are satisfied. Symbols X_{A_j}, $1 \leq j \leq r$, guarantee the correctness of the next step of the simulation. The work of agents R_D, $D \in N$, does not alter the sentential form, i.e. identical rewritings are applied. No other agent can be active in this phase of the derivation.
(4) In the fourth phase, agent R''_i applies the production labelled by i to substitute A with x. Furthermore, for any $D' \in N$ present in and any $D \in N$ absent from the sentential form agents R'_D and R_D should be activated, as well. Only in the case when $D' \in N$, $D' \neq C$ does the derivation terminate without the introduction of the trap symbol: R'_D and R_D rewrite $X_{D'}$ and X_D, respectively, to λ, the environment replaces X''_i with X and performs some identical rewritings, which completes the simulation of the production.

Should agent R_0 erase the nonterminal X in lieu of initiating the simulation of a production, then the derivation cannot produce any new strings. In such a case, if the sentential form is a terminal string, then it is an element of $L(G)$. Due to the construction of the simple eco-grammar system Γ, only words of $L(G)$ can be obtained. Thus our statement is verified for team mode $c^{=q}$, $q \in \mathcal{N}$.

To prove our claim for team mode $c^{\leq q}$, $q \in \mathcal{N}$, symbols D', X'_D and agents $R_{D'}$ should be omitted from the definition of Γ. R_D, $D \in N$, ought to be modified as follows:

$$R_D = \{X_D \to \chi, D \to \psi, Y_j \to F \mid 1 \leq j \leq q, \text{ where if } D \neq C \text{ then } \chi = \lambda, \psi = D, \text{ or if } D = C \text{ then } \chi = \psi = F\}.$$

The verification is analogous to that of team mode $c^{=q}$, $q \in \mathcal{N}$, consequently, it is left to the reader. □

Secondly, we focus on the $d^{=q}$ team derivation mode, $q \in \mathcal{N}$.

Theorem 13. *A language L over an alphabet T is recursively enumerable if and only if it can be obtained as $L = L' \cap T^*$ for some $L' \in EGL(d^{=q})$, $q \in \mathcal{N}$.*

Proof. Let us consider the context-free random context grammar of the same form as in the case of the previous proof. We will construct a simple eco–grammar system working in the $d^{=q}$ team mode, $q \in \mathcal{N}$, which simulates the derivation in G.

Let $X, F, X_k, X'_k, X''_k, Y_l, Y_{k,j}, Y'_{k,j}, Y''_{k,j}, 1 \leq k \leq n+1, 1 \leq l \leq q+1, 1 \leq j \leq q+4$, be new distinct symbols not in $N \cup T$. We define

$$\Gamma = (V, P_E, R_0^1, R_0^2, (R_k^i)_{1 \leq k \leq n, 1 \leq i \leq 6}, R_{n+1}^1, R_{n+1}^2, XY_1 \ldots Y_{q+1}S),$$

where $V = N \cup T \cup \{F, X\} \cup \{X_k, X'_k, X''_k, Y_{k,j}, Y'_{k,j}, Y''_{k,j} \mid 1 \leq k \leq n+1, 1 \leq j \leq q+4\} \cup \{Y_l \mid 1 \leq l \leq q+1\}$,

$$\begin{aligned} P_E = \{&a \to a \mid a \in N \cup T \cup \{F, X, X_k, X''_k \mid 1 \leq k \leq n+1\}\} \cup \{X'_k \to X''_k\} \\ &\{Y_{k,j} \to Y'_{k,j}, Y'_{k,j} \to Y''_{k,j}, Y''_{k,j} \to \lambda \mid 1 \leq k \leq n+1, 1 \leq j \leq q+4\} \cup \\ &\{Y_j \to \lambda \mid 1 \leq j \leq q+1\}. \end{aligned}$$

We also have

$$\begin{aligned} R_0^1 &= \{X \to X_k Y_{k,1} \ldots Y_{k,q+4}, Y_{q+1} \to F \mid 1 \leq k \leq n+1\}, \\ R_0^2 &= \{X \to F, Y_j \to \lambda \mid 1 \leq j \leq q+1\}, \\ R_{n+1}^1 &= \{X_{n+1} \to \lambda, Y_{n+1,j} \to F \mid q+1 \leq j \leq q+4\}, \\ R_{n+1}^2 &= \{X_{n+1} \to F, Y_{n+1,j} \to \lambda \mid 1 \leq j \leq q+4\}, \end{aligned}$$

and for each rule $(B, C) : A \to x$, labelled by k, $1 \leq k \leq n$, we introduce agents

$$\begin{aligned} R_k^1 &= \{X_k \to X'_k\}, \\ R_k^2 &= \{X_k \to F, \delta \to F, Y_{k,j} \to Y'_{k,j} \mid 1 \leq j \leq q-1 \text{ , where if } B \neq \emptyset, \\ &\qquad \text{then } \delta = B, \text{ or if } B = \emptyset \text{ then } \delta = A\}, \end{aligned}$$

$$\begin{aligned} R_k^3 &= \{C \to F, Y_{k,q+4} \to F\}, \\ R_k^4 &= \{X'_k \to X''_k, Y'_{k,j} \to Y''_{k,j} \mid 1 \leq j \leq q+1\}, \end{aligned}$$

$$\begin{aligned} R_k^5 &= \{X''_k \to F, A \to x, Y_{k,q+4} \to F, Y''_{k,q+4} \to F\}, \\ R_k^6 &= \{X''_k \to XY_1 \ldots Y_{q+1}, Y''_{k,j} \to F \mid 1 \leq j \leq q+2\}. \end{aligned}$$

We clarify how a derivation in G can be simulated by some derivation in Γ in the $d^{=q}$ team derivation mode, $q \in \mathcal{N}$. Assume that at some moment the current state of Γ is $XY_1 \ldots Y_{q+1}\omega$, where ω is a sentential form in G. Notice that the initial state of Γ is a string of this form. A derivation step in G, where the rule $(B, C) : A \to x$, labelled by k is to be employed may be simulated as follows:

(1) During the initial phase, only team $\mathcal{T}_0 = \{R_0^1, R_0^2\}$ can work. Agent R_0^1 indicates which rule is to be simulated by rewriting X to $X_k Y_{k,1} \dots Y_{k,q+4}$, $1 \leq k \leq n$. This component may also terminate the simulation process through the substitution of X for $X_{n+1} Y_{n+1,1} \dots Y_{n+1,q+4}$. Component R_0^2 acts in parallel with R_0^1, R_0^2 erases one of the markers Y_j, $1 \leq j \leq q+1$. The other Y_ls, $1 \leq l \leq q+1, l \neq j$, will be deleted by the environment.
(2) At the second stage, the simulation is continued by a team formed from $\mathcal{T}_1 = \{R_k^1, R_k^2\}$. This team can be formed and used if and only if symbol B (or symbol A if $B = \emptyset$) occurs in the sentential form.
(3) During the third phase, the team that can be activated is $\mathcal{T}_2 = \{R_k^3, R_k^4\}$, which checks the absence of C in the sentential form. If C is not present in the sentential form, then the environment continues the derivation.
(4) Finally, the team $\mathcal{T}_3 = \{R_k^5, R_k^6\}$ is employed to execute rule $A \to x$ and to introduce nonterminals $X, Y_1, \dots, Y_{q+1}$, whereas the other markers are removed by the developmental rules of the environment.

When team $\mathcal{T}_4 = \{R_{n+1}^1, R_{n+1}^2\}$ is activated, all markers are removed, thus none of the agents of Γ can be applied anymore. Owing to the construction of the simple eco-grammar system Γ, only words of $L(G)$ can be produced. Thus our statement is proved for the $d^{=q}$ team mode, $q \in \mathcal{N}$. □

Lastly, we deal with team derivation modes $t^{=V_B}$ and $t^{\supseteq V_B}$, $\emptyset \neq V_B \subseteq V$.

Theorem 14. *For any recursively enumerable language $L \subseteq T^*$, there exists a simple eco–grammar system $\Gamma = (V, P_E, R_1, \dots, R_m, \omega_E)$, $m \geq 1$, and V_B, $\emptyset \neq V_B \subseteq V$, such that $L = L' \cap T^*$ holds, where $L' \in EGL(\alpha)$, $\alpha \in \{t^{=V_B}, t^{\supseteq V_B}\}$.*

Proof. Let us consider the context–free random context grammar G of the same form as in the previous proofs. Let us construct a simple eco–grammar system, which simulates a derivation in G. Let X, F, X_i, X_i', X_i'', $1 \leq i \leq n$, X_{n+1}, be new distinct symbols not in $N \cup T$. The underlying simple eco–grammar system is as follows:

$$\Gamma = (V, P_E, R_0, R_{n+1}, (R_{i,j})_{1 \leq i \leq n, 1 \leq j \leq 7}, XS),$$

where

$$\begin{aligned} V &= N \cup T \cup \{X, F\} \cup \{X_i, X_i', X_i'' \mid 1 \leq i \leq n\} \cup \{X_{n+1}\}, \\ V_B &= N \cup \{X, X_i, X_i', X_i'' \mid 1 \leq i \leq n\} \cup \{X_{n+1}\}, \\ P_E &= \{a \to a \mid a \in V\}, \end{aligned}$$

and

$$\begin{aligned} R_0 &= \{X \to X_i \mid 1 \leq i \leq n+1\} \cup \{D \to D \mid D \in N\}, \\ R_{n+1} &= \{X_{n+1} \to \lambda\}. \end{aligned}$$

Moreover, for each rule $(B, C) : A \rightarrow x$, labelled by i, $1 \leq i \leq n$, we introduce agents

$$\begin{aligned}
R_{i,1} &= \{\delta \rightarrow F, X_i \rightarrow X_i', D \rightarrow F \mid D \in N \setminus \{\delta\}, \text{ where if } B \neq \emptyset, \\
&\qquad \text{then } \delta = B, \text{ or if } B = \emptyset \text{ then } \delta = A\}, \\
R_{i,2} &= \{\delta \rightarrow \delta, X_i \rightarrow F, D \rightarrow F \mid D \in N \setminus \{\delta\}, \text{ where if } B \neq \emptyset, \\
&\qquad \text{then } \delta = B, \text{ or if } B = \emptyset \text{ then } \delta = A\}, \\
R_{i,3} &= \{X_i' \rightarrow F, D \rightarrow F, C \rightarrow F \mid D \in N \setminus \{C\}\}, \\
R_{i,4} &= \{X_i' \rightarrow X_i'', D \rightarrow F, C \rightarrow F \mid D \in N \setminus \{C\}\}, \\
R_{i,5} &= \{X_i' \rightarrow F, D \rightarrow D, C \rightarrow F \mid D \in N \setminus \{C\}\}, \\
R_{i,6} &= \{X_i'' \rightarrow X, D \rightarrow F \mid D \in N\}, \\
R_{i,7} &= \{X_i'' \rightarrow F, A \rightarrow x, D \rightarrow F \mid D \in N \setminus \{A\}\}.
\end{aligned}$$

We demonstrate how an arbitrary derivation in G can be simulated by a derivation in Γ applying analogous considerations to the ones of the proof of the previous two statements. Suppose that at some moment the current state of Γ is $X\omega$, where ω is a sentential form in G. Indeed, the initial state of Γ is of this form. A derivation step in G during which the rule $(B, C) : A \rightarrow x$ labelled by i is employed, can be simulated as follows:

(1) At the first step, only component R_0 can work. It indicates the rule to be simulated: it rewrites X to X_i, $1 \leq i \leq n$. Observe that this component may also terminate the simulation process through the substitution of X with X_{n+1}.
(2) At the second stage, the team consisting of $R_{i,1}$ and $R_{i,2}$ continue the simulation. The simulation works if and only if $R_{i,1}$ uses the rule $X_i \rightarrow X_i'$ and $R_{i,2}$ the rule $B \rightarrow B$ (or $A \rightarrow A$ if $B = \emptyset$). In this way we can check whether the permitting context condition is satisfied.
(3) During the third phase, we examine the fulfillment of the forbidding context condition. In the third phase, either the team consisting of agents $R_{i,3}, R_{i,4}$ and $R_{i,5}$ or the team composed of agents $R_{i,4}$ and $R_{i,5}$ may be activated, which depends on the presence of C in the sentential form. If symbol C occurs in the sentential form, then the joint action of agents $R_{i,3}, R_{i,4}$ and $R_{i,5}$ introduces the trap symbol and the derivation cannot be continued. If C is absent from the sentential form, then $R_{i,4}$ rewrites X_i' to X_i'' and $R_{i,5}$ performs an identical rewriting step on one of the nonterminals.
(4) If the forbidding context condition is satisfied, then at the fourth stage the team formed by $R_{i,6}$ and $R_{i,7}$ is activated and it finishes the simulation of production i through the replacement of an occurrence of A with x and X_i'' with X.

If symbol X is rewritten to X_{n+1}, R_{n+1} erases X_{n+1} from the sentential form. None of the agents can be employed anymore, after R_{n+1} has been activated, since all markers have been removed from the environmental string. Due to the construction of the simple eco-grammar system Γ, only words of $L(G)$ can be generated.

Consequently, our assertion is verified for team derivation modes $t^{=V_B}$ and $t^{\supseteq V_B}$, $\emptyset \neq V_B \subseteq V$. □

7. Discussion

In this article motivated by the bottom–up–clustering algorithm [2, 7, 16, 22] we have extended the conditions of dynamic team constitution [13] in simple eco–grammar systems [11]. The relationships of simple eco–grammar systems formed according to the newly introduced conditions to each other have been established. It can be concluded that in certain team derivation modes the number of agents plays a crucial part in cluster formation. It has been investigated how simple eco–grammar systems consisting of dynamically formed teams given various constitution modes are related to the language classes of the Chomsky hierarchy. From the language classes these systems are capable of generating, we can deduce the difficulty of the problem they can solve. We have had to impose various restrictions on the alphabet of finite languages so that we are able to generate them by simple eco–grammar systems. The cooperation of teams has been shown to lead to quite a large computational power.

The model that we have presented in this paper is purely a syntactical one. In the sequel, we compare our approach with the approaches of contemporary research and propose some further research directions.

First, bottom–up–clustering is a way of extracting community structure from the network [26]. The extraction is based on the measure of the connection strength. As a result of the clustering process, the network will be divided into densely connected subgraphs, i.e. clusters. The communication between nodes belonging to different subgraphs is minimized. In our work, we have regarded the connection strength as the difference in the level of competence/excitation of the agents. The communication of the agents can be viewed as the emergence of their intensive interaction with their commonly shared environment [10]. The communication between two teams is minimized, since at a given instant only the agents of the same team are allowed to have an impact on the environmental string. Our approach differs in some aspects from the approaches presented in the literature [9, 26]. On the one hand, in our work, the communication occurs in an indirect manner. The agents perform computations and include the results in the environmental string. On the other hand, in the bottom–up–clustering algorithm teams can work in a parallel manner [9, 26], which implies that the concept of teams of a team (subclusters of a cluster [9]) may be elaborated in our mathematical model, hence the potential of construction of a hierarchical structure.

Secondly, the amelioration of the performance of the bottom–up–clustering algorithm is investigated from two different points of view in the literature [2, 7, 8, 16, 20, 21]. It is questionable whether the modifications incorporated into the bottom–up–clustering algorithm produce better answers than previous solutions (according to well–defined quality measure) or the same answers, in less time (theoretically or in practice). In our model, the goodness of the answer corresponds to the largeness of the language class that can be obtained due to the collaboration of the agents in various team modes. The number of time steps can be measured in terms of derivation steps, if we assume that one time step is equal to one team derivation step.

Finally, other team derivation modes can be introduced. The difference in the level of competence/excitation of the agents may be modified over time. Hybrid team derivation modes [6, 17, 18, 24, 27], i.e. the employment of the different conditions of dynamic team constitution conditions at various derivation steps or the combination of different conditions of dynamic team constitution conditions by means of logical operators, constitute a subject of further research.

References

[1] A. L. Barabási, R. Albert and H. Jeong, "Scale–Free Characteristics of Random Networks: the Topology of the World–Wide Web," *Physica A* **281** (2000) 69–77.
[2] R. Baeza–Yates and B. Ribeiro–Neto, *Modern Information Retrieval* (Addision Wesley, 1999).
[3] M. H. ter Beek, "Teams in Grammar Systems: Hybridity and Weak Rewriting," *Acta Cybernetica* **12** (1996) 427–444.
[4] M. H. ter Beek, "Simple Eco–Grammar Systems with Prescribed Teams," in *Grammatical Models of Multi–Agent Systems*, eds. Gh. Păun and A. Salomaa, (Gordon and Breach, 1999) pp. 113–135.
[5] B. Bollobás, *Random Graphs* (Academic, London, 1985).
[6] H. Bordihn and M. Holzer, "Grammar Systems with Negated Conditions in Their Cooperation Protocols," *Journal of Universal Computer Science* **6(12)** (2000) 1165–1184.
[7] S. Chakrabarti, *Mining the Web* (Morgan Kaufmann, 2003).
[8] J. Cong and S. K. Lim, "Edge Separability Based Circuit Clustering with Application to Circuit Partitioning," UCLA Computer Science Technical Report, #990030.
[9] J. Cong and M. Smith, "A Parallel Bottom–Up Clustering Algorithm with Applications to Circuit Partitioning in VLSI Design," *Design Automation Conference* 1993 pp. 755–760. `http://citeseer.ist.psu.edu/cong93parallel.html`
[10] E. Csuhaj–Varjú, J. Dassow, J. Kelemen and Gh. Păun, *Grammar Systems: A Grammatical Approach on Distribution and Cooperation* (Gordon and Breach, London, 1994).
[11] E. Csuhaj–Varjú, J. Kelemen, A. Kelemenová and Gh. Păun, "Eco–Grammar Systems. A Grammatical Framework for Studying Lifelike Interactions," *Articial Life* **3** (1997) 1–28.
[12] E. Csuhaj–Varjú and A. Kelemenová, "Team Behaviour in Eco–Grammar Systems," *Theoretical Computer Science* **209** (1998) 213–224.
[13] E. Csuhaj–Varjú and V. Mitrana, "Dynamical Teams in Eco–Grammar Systems," *Fundamenta Informaticae* **44** (2000) 83–94.
[14] J. Dassow and J. Kelemen, "On Parallelism in Colonies," *Cybernetics and Systems: An International Journal* **24** (1993) 37–49.
[15] J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory* (Springer–Verlag, New York, 1996).
[16] D. Eppstein, "Fast Hierarchical Clustering via Dynamic Closest Pairs,"

Dept. Information and Computer Science Univ. of California, Irvine, http://www.ics.uci.edu/~eppstein/
[17] H. Fernau, M. Holzer and R. Freund, "Hybrid Modes in Cooperating Distributed Grammar Systems: Internal versus External Hybridization," *Theoretical Computer Science* **259** (2001) 405-426.
[18] H. Fernau, M. Holzer and R. Freund, "Hybrid Modes in Cooperating Distributed Grammar Systems: Combining the t–Mode with the Modes $\leq k$ and $= k$," *Theoretical Computer Science* **299** (2003) 633-662.
[19] *Handbook of Formal Languages*, eds. G. Rozenberg and A. Salomaa, (Springer, Berlin, 1997).
[20] C. Hung, S. Wermter and P. Smith, "Predictive Top–Down Knowledge Improves Neural Exploratory Bottom–Up Clustering," *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 2997, 2004.
[21] A. Lőrincz, B. Gábor, S. Mandusitz and Zs. Palotai, "Bottom–Up Clustering and Top–Down Shattering of Scale–Free Environments for Information Fusion," *Information Fusion 2004*, 28 June – 1 July, 2004, Stockholm, Sweden. Available on: http://www.fusion2004.foi.se/papers/IF04-0471.pdf
[22] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, 2003).
[23] M. Marchioria and V. Latora, "Harmony in the Small–World," *Physica A* **285** (2000) 39–56.
[24] V. Mitrana, "Hybrid Cooperating/Distributed Grammar Systems," *Comput. Artif. Intell.* **12(1)** (1993) 83–88.
[25] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review* **45** (2003) 167-256.
[26] M. E. J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Phys. Rev. E* **69** (2004) 026113.
[27] Gh. Păun, "On the Generative Capacity of Hybrid CD Grammar Systems," *J. Inform. Process. Cybernet.* **EIK 30(4)** (1994) 231–244.
[28] Gh. Păun and G. Rozenberg, "Prescribed Teams of Grammars," *Acta Informatica* **31** (1994) 525–537.
[29] G. Rozenberg and A. Salomaa, *Mathematical Theory of L systems* (Academic Press, New York, 1980).
[30] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[31] D. Wätjen, "Function–Dependent Teams in Eco–Grammar Systems," *Theoretical Computer Science* **306** (2003) 39–53.
[32] D. Wätjen, "Remarks on Simple Eco–Grammar Systems with Prescribed Teams," *Journal of Automata, Languages and Combinatorics* **8(1)** (2003) 85–108.
[33] D. J. Watts and S. H. Strogatz, "Collective Dynamics of Small–World Networks," *Nature* **393** (1998) 440–442.