Bounded life recources in colonies

Alica Kelemenová
Institute of Computer Science
Silesian University
Bezručovo nám. 13, 746 01 Opava
Czech Republic
e-mail: kelemenova@fpf.slu.cz

Lecture Notes
Tarragona, March 11-15, 2002

1. Introduction

Colonies as language generating devices rooted in the theory of grammar systems [3, 6] have been introduced in [9] and developed during the nineties in several directions, see, e.g., [1, 4, 7, 10, 11, 14, 15, 16, 19, 20].

The basic idea behind an attempt to introduce colonies was to create exact methods to study emergency, using theory of formal grammars and languages. The goal was to build up from "as simple as possible" components cooperating in an elementary level a system which increases the generative power of the components.

In order to preserve the communication between components of a colony as simple as possible, one can look for some regulations of the component participation in the process of string generation. Time delays associated with components have been invented already in [9]; see also [10].

In this overview paper we discuss limitations of the resources activation (components, productions, nonterminals) of colonies.
Four variants of colonies with limited activation of components are studied, namely colonies with delays, colonies with increasing activity of components, colonies with bounded-life components and colonies with bounded-frequency components. The generative power of these colonies is discussed. Some properties of a specific descriptional complexity measure, namely the number of immortal components of a colony with bounded-life components, are also discussed.

2. Colonies

The basic idea behind an attempt to introduce colonies was to create exact methods to study emergency, using theory of formal grammars and languages. The goal was to build up from "as simple as possible" components cooperating in an elementary level a system which increases the generative power of the components. By an *elementary* cooperation of components we understand the blackboard like cooperation where components act on/cooperate through a common tape which is the only form of their interactive space. The main point is that the action of each component *depends only on the situation on the interactive tape* and possibly on the inner conditions (inner state) of the component. It is *independent on the inner conditions of any other component*. It turned out that it is sufficient to deal with components with finite behaviour, i.e. components, which produce finite languages have the above required property. A schematic view of a colony is given in Fig. 1.

A colony (of grammars) was introduced as a collection of grammars producing finite languages, which operate sequentially on a common string forming their interactive space in [9]. By sequential colonies operating in the simplest way (i.e. in each time period a component rewrites one occurrence of the start symbol) exactly all context free languages can be formed [9]. Complex view to the generative power of sequential colonies is given in [11].

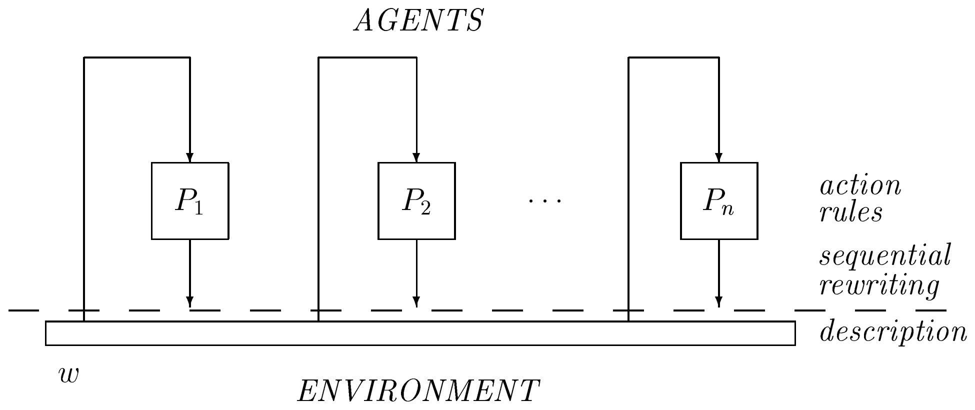

Fig.1: A schematic view of a colony

Before we shall deal with extensions we recall the definition of the basic variant of a colony:

Definition 2.1: *A colony* $\mathcal{C}$ is a construct $\mathcal{C} = (V, T, \mathcal{R})$, where

(i) V, $T \subseteq V$ are *total and terminal alphabets* of the colony, respectively, and

(ii) $\mathcal{R} = \{(S_i, F_i)\}_{i=1}^{i=n}$ is a finite sequence of grammars each of them represented by the starting symbol $S_i \in V$ and by the finite language $F_i \subset (V - \{S_i\})^*$.

A pair (S_i, F_i) denotes *the i-th component* of $\mathcal{C}$. A component (S_i, F_i) with $F_i = \{w_1, \ldots, w_{n_i}\}$ will be given also in the form $(S_i \rightarrow w_1 | \ldots | w_{n_i})$.

An activity of a colony is realized by sequential action of components in a given string. Elementary changes of strings, where one of the components replaces one letter S_i of a string by a word z from F_i is called *basic* derivation step and denoted by $\Longrightarrow_{\mathcal{C}}$. Formally:

Definition 2.2: For $x, y \in V^*$ we write $x \Longrightarrow_{\mathcal{C}} y$ iff for some i, $1 \leq i \leq n$, $x = x_1 S_i x_2$, $y = x_1 z x_2$ and $z \in F_i$.
$\Longrightarrow_{\mathcal{C}}^*$ is the reflexive and transitive closure of $\Longrightarrow_{\mathcal{C}}$.

Definition 2.3: The *language* determined by a colony $\mathcal{C}$ starting with the axiom $w_0 \in V^*$ is given by

$$L(\mathcal{C}, w_0) = \{v |\ w_0 \Longrightarrow^* v,\ v \in T^*\}.$$

We denote by COL the collection of languages defined by colonies.

Theorem 2.1 [9]: COL is identical with the set of all context free languages.

Example 2.1: Consider the colony $\mathcal{C} = (V, T, \mathcal{R})$, where
$V = \{A, B, A_1, B_1\} \cup T$,
$T = \{a, b, c\}$,
$\mathcal{R} = \{(A \to aA_1b), (B \to cB_1|c), (A_1 \to ab|A), (B_1 \to B|c)\}$.

Then the behaviour of the colony $\mathcal{C}$ for the axiom AB is the context free language

$$L(\mathcal{C}, AB) = \{a^n b^n c^m | m, n \geq 1\}.$$

An example of derivation in the colony is
$AB \Longrightarrow aA_1bB \Longrightarrow aA_1bc \Longrightarrow aabbc.$

3. Colony with delays

There are some motivation to introduce regulations to the frequency of possible acting of the components. One can study components sensitive to the time period past from their last activity or components sensitive to the amount of previously executed steps. Following are examples of such properties:

- after an action the component cannot act during some predetermined time period (interpreted as "it needs rest", "it needs to renew energy"). That case, a colony with constant delays was introduced in [11].
- the activity of the component decreases/increases with the number of actions it had already executed ("exhausting of the energy", "learned behaviour", ...). That case, a colony with nonconstant delays was introduced in [11].

A colony with delays models the situation when each component has to postpone its action by a given number of steps after its last action. Delays for different components can be different. Eventually, it can be dependent on the number of activation of the component during the ongoing derivation. Delays of components are represented by (possibly constant) functions defined on natural numbers.

First, we introduce a colony with constant delays.

Definition 3.1: A *colony with constant delays* is a 3-tuple $\mathcal{C}_d = (V, T, \mathcal{R})$, where V, $T \subseteq V$ are *total and terminal alphabets* of the colony, respectively, and $\mathcal{R} = \{(S_i, F_i, d_i)\}_{i=1}^{i=n}$ is a finite set of grammars, each associated with the starting symbol $S_i \in V$, finite language $F_i \subset (V - \{S_i\})^*$ and natural number d_i.

In the present model a fully reactive action of $i-$th component is postponed for d_i steps after its last activation. To describe the derivation we have to count an actual delay for every component, therefore a derivation step of a colony with delays is the relation between configurations, which are of the form $\langle w, t_1, t_2, \ldots, t_n \rangle$, where w is a string over V^* and $t_1, t_2, \ldots, t_n$ are of nonnegative integers representing time period during which the corresponding component has to be nonactive. $t_i = 0$ corresponds to a potentially active component i of the colony.

For derivation step of a colony $\mathcal{C}_d$ with constant delays we have:

Definition 3.2: Let $\langle w_1, t_1, \ldots, t_n \rangle$ and $\langle w_2, t'_1, \ldots, t'_n, \rangle$ be configurations of a colony with constant delays. We write
$\langle w_1, t_1, \ldots, t_n \rangle \Longrightarrow_{\mathcal{C}_d} \langle w_2, t'_1, \ldots, t'_n \rangle$ iff

- there is j, $1 \leq j \leq n$ such that $t_j = 0$ and $w_1 = x_1 S_j x_2$

- $w_2 = x_1 z x_2$ for some $z \in F_j$,
- $t'_j = d_j$ and t'_i =max$\{0, t_i - 1\}$ for all i, $i \neq j$.

Note: A derivation step of a colony without delay corresponds to zero delay of all components in every configuration.

Definition 3.3: The *language* defined by a colony $\mathcal{C}_d$ with axiom w_0 and with initial delays of components given by $t_i, 1 \leq i \leq n$ is the set

$$L(\mathcal{C}_d, w_0, t_1, \ldots t_n) = \{w \in T^* : \langle w_0, t_1, \ldots t_n \rangle \Longrightarrow^*_{\mathcal{C}_d} \langle w, t'_1, \ldots, t'_n \rangle\}.$$

We denote by COL_d the collection of all languages defined by the colonies with constant delay.

Example 3.1: Consider the colony $\mathcal{C}_d = (V, T, \mathcal{R})$, where
$V = \{A, B, A_1, B_1\} \cup T$,
$T = \{a, b, c\}$,
$\mathcal{R} = \{(A \to aA_1b, 3), (B \to cB_1|c, 3), (A_1 \to ab|A, 3), (B_1 \to B|c, 3)\}$.

Then the behaviour of the colony for the axiom AB and starting delay of components being 0,1,2 and 3, respectively, is the context sensitive language

$$L_{abc} = L(\mathcal{C}_d, AB, (0, 1, 2, 3)) = \{a^n b^n c^n | n \geq 2\}.$$

An example of derivation in the colony is
$\langle AB, 0, 1, 2, 3 \rangle \Longrightarrow \langle aA_1bB, 3, 0, 1, 2 \rangle \Longrightarrow \langle aA_1bcB_1, 2, 3, 0, 1 \rangle$
$\Longrightarrow \langle aabbcB_1, 1, 2, 3, 0 \rangle \Longrightarrow \langle aabbcc, 0, 1, 2, 3 \rangle$.

Theorem 3.1: $COL \subset COL_d$.
Proof: The inclusion follows directly from the definition and for the language L_{abc} from Example 2.1.1 we have $L_{abc} \in COL_d - COL$.

Theorem 3.2: The set of Parikh vectors of the language L for $L \in COL_d$ is semilinear.
Sketch of the proof: It can be done in the similar way as analogical theorem for context free languages.

Theorem 3.3 [21]: COL_d is closed under union, mirror image and intersection. COL_d not closed under catenation and iteration.

We extend the model to the case, where the delay of components of a colony is not necessarily constant. We assume that the delay of i-th component is determined by function f_i, which associates to a natural number j (representing j-th activation of the component) a delay $f_i(j)$.

Definition 3.4: A *colony with delay* is a structure $\mathcal{C}_f = (V, T, \mathcal{R})$, where

- V, T have same meaning as in a colony, and
- $\mathcal{R} = \{(S_i, F_i, f_i)\}_{i=1}^{i=n}$ is a finite sequence of grammars, each associated with delay functions $f_i : \mathcal{N} \to \mathcal{N}$, where $\mathcal{N}$ is the set of nonnegative integers.

Similarly as in the previous case activity of components is described by a derivation step. In the this case, delays are functions depending on the number of activations of each components. Therefore a configuration in this case contains a pair $\langle t_i, j_i \rangle$ for each component, where t_i represents actual number of derivation steps during which the component is not allowed to act and j_i is the number of already finished activations of the i-th component. The i-th component can be active for $t_i = 0$ only and after executing the derivation step the value of t_i becomes $f_i(j_i)$ and the value of activations of the i-th component (j_i) increases by 1.

Definition 3.5: Let $\langle w_1, \langle t_1, i_1 \rangle, \ldots, \langle t_n, i_n \rangle \rangle$ and $\langle w_2, \langle t'_1, i'_1 \rangle, \ldots, \langle t'_n, i'_n \rangle \rangle$ are configurations of a colony with nonconstant delay. We write
$\langle w_1, \langle t_1, i_1 \rangle, \ldots, \langle t_n, i_n \rangle \rangle \Longrightarrow_{\mathcal{C}_f} \langle w_2, \langle t'_1, i'_1 \rangle, \ldots, \langle t'_n, i'_n \rangle \rangle$ iff

- $t_j = 0$ and $w_1 = x_1 S_j x_2$ for some j, $1 \leq j \leq n$ and
- $w_2 = x_1 z x_2$ for some $z \in F_j$,
- $t'_j = f_j(i_j), i'_j = i_j + 1$ and
- $t'_k = \max\{0, t_k - 1\}$, $i'_k = i_k$ for every k, $k \neq j$.

Note: In an initial configuration we have $i_1 = i_2 = \ldots = i_n = 0$.
For the components with constant delay function we shall write simply t_j instead of $\langle t_j, i_j \rangle$.

Definition 3.6 The *language* defined for a colony $\mathcal{C}_f$ with axiom w_0 and with an initial delay of i-th component $\langle t_i, 0 \rangle$ is the set

$$L(\mathcal{C}_f, w_0, \langle t_1, 0 \rangle, \langle t_2, 0 \rangle \ldots, \langle t_n, 0 \rangle) =$$

$$\{w \in T^* : \langle w_0, \langle t_1, 0 \rangle, \langle t_2, 0 \rangle \ldots, \langle t_n, 0 \rangle \rangle \Longrightarrow^*_{\mathcal{C}_f} \langle w, \langle t'_1, i'_1 \rangle, \ldots, \langle t'_n, i'_n \rangle \rangle\}.$$

We denote by COL_f the collection of languages defined by the colonies with non constant delay.

Example 3.2: Consider the colony $\mathcal{C}_f = (V, T, \mathcal{R})$, where
$V = \{S, S_1, S_2\} \cup T$,
$T = \{a, b\}$,
$\mathcal{R} = \{(S \to aS_1, 1), (S \to bS_2 | b, 2^{p+1} + 1), (S_1 \to S, 1), (S_2 \to S, 1)\}$.
We determine the behaviour of the colony for the axiom S and initial delays $1, \langle 0, 0 \rangle, 0$ and 0, respectively.

At the beginning the derivation produces a letter b or proceeds as follows
$\langle S, 1, \langle 0, 0 \rangle, 0, 0 \rangle \Longrightarrow \langle bS_2, 0, \langle 3, 1 \rangle, 0, 0 \rangle \Longrightarrow \langle bS, 0, \langle 2, 1 \rangle, 0, 1 \rangle \Longrightarrow$
$\langle baS_1, 1, \langle 1, 1 \rangle, 0, 0 \rangle \Longrightarrow \langle baS, 0, \langle 0, 1 \rangle, 1, 0 \rangle$.

In the next step we have three possibilities to continue:
(1) to produce aS_1 by the first component
$\langle baaS_1, 1, \langle 0, 1 \rangle, 0, 0 \rangle$
(2) to produce b by the second component and to end the derivation
$\langle bab, 0, \langle 7, 2 \rangle, 0, 0 \rangle$
(3) to produce bS_2 by the second component
$\langle babS_2, 0, \langle 7, 2 \rangle, 0, 0 \rangle$.

Generally, the second component is active for a configuration
$\langle ba^{p_0}b \ldots ba^{p_i}S, 0, \langle 0, i \rangle, x, y \rangle$, $p_i \geq 2^i$, $x = 1$ or $y = 1$ therefore

$$L(\mathcal{C}_f, S, 1, \langle 0, 0 \rangle, 0, 0) = \{ba^{p_0}ba^{p_1} \ldots ba^{p_j}b | \ \ p_i \geq 2^i, \ \ i \leq j\},$$

which is the context sensitive not context free language.

Theorem 3.4 [21]: $COL_d \subset COL_f$
Proof: The inclusion follows directly from the definition. For the language L from Example 2.1.2 by Theorem 2.1.2 we have $L \in COL_f - COL_d$.

Theorem 3.5 [21]: COL_f is closed under union and mirror image. COL_f is not closed under concanation, iteration and intersection.

4. Colony with time increasing activities

The activity of the component increases with the time passed from its last action ("it need food more urgently", "it accumulates more and more energy"). That case, a colony with time increasing activity was introduced in [11].

The activity of the components can increase with the time passed from their last action ("they need food more urgently", "they accumulate more and more energy"). To define colony with such a behaviour the model differs from the basic form of colony in the definition of derivation step only. In accordance with the motivation, during the derivation process described by relations on configurations $\langle w, t_1, \ldots, t_n \rangle$ t_i becomes 0 for the actually working component and all other components increase their value t_i by 1. A component with the greatest value t_j having its starting symbol in the current word is allowed to act (it is the best one, the most quick one.)

Definition 4.1: A *colony with time increasing activity* is a colony with standard three-tuple $\mathcal{C}_{max} = (V, T, \mathcal{R})$.

A derivation step for colonies with time increasing activity is defined for configurations $\langle w, t_1, \ldots, t_n \rangle$, where $w \in V^*$ and $t_i, 1 \leq i \leq n$ are numbers.

Definition 4.2: Let $\langle w_1, t_1, \ldots, t_n, \rangle$ and $\langle w_2, t'_1, \ldots, t'_n, \rangle$ are configurations of a colony with time increasing activity. We write
$\langle w, t_1, \ldots, t_n \rangle \Longrightarrow_{\mathcal{C}_{max}} \langle w', t'_1, \ldots, t'_n) \rangle$ iff

- $w = x_1 S_j x_2$, $t_j = max\{t_i : \ S_i \text{ is a letter of } w\}$ for some j, $1 \leq j \leq n$ and

- $w' = x_1 z x_2$ for some $z \in F_j$,
- $t'_j = 0$ and $t'_i = t_i + 1$ for $i \neq j$.

Definition 4.3: *Language* defined by a colony $\mathcal{C}_{max}$ with axiom w_0 and the initial time t_i is the set

$$L(\mathcal{C}_{max}, w_0, t_1, \ldots, t_n) = \{w \in T^* : \langle w_0, t_1, \ldots, t_n \rangle \Longrightarrow^*_{\mathcal{C}_{max}} \langle w, t'_1, \ldots, t'_n \rangle\}.$$

We denote by $COL_{\mathcal{I}}$ the collection of languages defined by the colonies with time increasing activity.

Example 4.1: Consider the colony $\mathcal{C}_{max} = (V, T, \mathcal{R})$, where
$V = \{A, B, C, a, b, c\}$,
$T = \{a, b, c\}$,
$\mathcal{R} = \{(a \to A), (b \to B), (c \to C), (A \to aa), (B \to bb), (C \to cc)\}$.

We determine the behaviour of the colony for the axiom abc initial values $t_1 = 5, t_2 = 4, t_3 = 3, t_4 = 2, t_5 = 1$.

$\langle abc, 5, 4, 3, 2, 1, 0 \rangle \Longrightarrow \langle Abc, 0, 5, 4, 3, 2, 1 \rangle \Longrightarrow \langle ABc, 1, 0, 5, 4, 3, 2 \rangle \Longrightarrow$
$\langle ABC, 2, 1, 0, 5, 4, 3 \rangle \Longrightarrow \langle aaBC, 3, 2, 1, 0, 5, 4 \rangle \Longrightarrow$
$\langle aabbC, 4, 3, 2, 1, 0, 5 \rangle \Longrightarrow \langle aabbcc, 5, 4, 3, 2, 1 \rangle$.

$L_{abc} = L(\mathcal{C}_{\mathcal{I}}, abc, (5, 4, 3, 2, 1, 0)) = \{a^i b^i c^i : \quad i \geq 1\}$.

Example 4.2: Consider the colony $\mathcal{C}_{max} = (V, T, \mathcal{R})$, where
$V = \{A, A_1, A_2, B\} \cup T$,
$T = \{a, b\}$,
$\mathcal{R} = \{(A \to aA_1), (A_1 \to A), (A \to Bb|A_2 b), (A_2 \to A), (B \to ab)\}$.

We determine the behaviour of the colony for the axiom AA and initial times are $3, 2, 0, 0$ and 0, respectively.

$\langle AA, 3, 2, 0, 0, 0 \rangle \Longrightarrow \langle aA_1A, 0, 3, 1, 1, 1 \rangle \Longrightarrow \langle aAA, 0, 0, 2, 2, 2 \rangle \Longrightarrow$
$\langle aAA_2b, 2, 1, 0, 3, 3 \rangle \Longrightarrow \langle aAAb, 3, 2, 1, 0, 4 \rangle \Longrightarrow$
$\langle aaA_1Ab, 0, 3, 2, 1, 5 \rangle \Longrightarrow \langle aaAAAb, 1, 0, 3, 2, 6 \rangle \Longrightarrow$

$\langle aaBbAb, 2, 1, 0, 3, 7\rangle \Longrightarrow \langle aaabbAb, 3, 2, 1, 4, 0\rangle \Longrightarrow$
$\langle aaabbaA_1b, 0, 3, 2, 5, 1\rangle \Longrightarrow \langle aaabbaAb, 1, 0, 3, 6, 2\rangle \Longrightarrow$
$\langle aaabbaBbb, 2, 1, 0, 7, 3\rangle \Longrightarrow \langle aaabbaabbb, 3, 2, 1, 8, 0\rangle.$

$L(\mathcal{C}_{\mathcal{I}}, AA, (3, 2, 0, 0, 0)) = \{a^{i_1}b^{i_2}a^{j_1}b^{j_2} : \quad i_1 + j_1 = i_2 + j_2 \geq 2\}.$

Theorem 4.1 [21]: $COL_b \subset COL_{\mathcal{I}}$.
Proof: The inclusion follows directly from the definitions and for the language L_{abc} from Example 2.2.1 we have $L_{abc} \in COL_{\mathcal{I}} - COL_b$.

Theorem 4.2 [21]: $COL_{\mathcal{I}}$ is closed under union and mirror image. COL_f is not closed under concanation, iteration and intersection.

5. Colony with bounded-life components

In the present section we discuss a limitation with respect to the number of component activations. We associate a possible infinite natural number with each component of a colony which restricts its possibility to participate arbitrarily many times in the string generation. This number may be viewed as the lifetime of that component. Components with unbounded participation in the generation of strings are called immortal. This type of regulation seems to be very natural in view of the aforementioned motivation. Obviously, each society of reactive agents contains agents which cannot act arbitrarily many times by different reasons.

Some components, having a finite lifetime, are called *bounded-life* components and others – the so called *immortal* ones – can be used arbitrarily many times.

Definition 5.1: A *colony with bounded-life components* is a structure

$$\mathcal{C} = (V, T, (S_1, F_1), \ldots, (S_n, F_n), w, bl),$$

where

- $(V, T, (S_1, F_1), \ldots, (S_n, F_n), w)$ is a usual colony, which will be denoted by $\bar{\mathcal{C}}$,

- bl is a mapping from the set $\{1, 2, \ldots, n\}$ into the set $\mathcal{N} \cup \{\infty\}$, which associates to each component of $\mathcal{C}$ a natural number representing the lifetime of that component. If $bl(r) = \infty$, then we say that the component r is *immortal.* The lifetime of a component indicates the maximal number of applications of that component in the process of changing the environment.

Definition 5.2: For every derivation D

$$y \Longrightarrow_{i_1} x_1 \Longrightarrow_{i_2} x_2 \ldots \Longrightarrow_{i_p} x_p = x,$$

in $\mathcal{C}$ and for every r, $1 \leq r \leq n$, we define

$$K_r(y, D, x) = \mathrm{card}(\{j | i_j = r, 1 \leq j \leq p\}).$$

Moreover, we denote by $|D|$ the length of the derivation D, that is $|D| = p$. For every word $x \in L(\bar{\mathcal{C}})$ we define

$$K_r(x, \mathcal{C}) = \min\{K_r(w, D, x) \mid D \text{ is a derivation of } x \text{ in } \mathcal{C}\}.$$

The *viable behavior* of $\mathcal{C}$ is defined by

$$L_l(\mathcal{C}) = \{x \in L(\bar{\mathcal{C}}) \mid K_r(x, \mathcal{C}) \leq bl(r), \text{ for all } 1 \leq r \leq n\}.$$

First we prove that this regulation does not lead to an increase in the computational power.

Theorem 5.1: For a given colony with bounded-life components $\mathcal{C}$ one can construct a colony $\mathcal{C}'$ with $L_l(\mathcal{C}) = \mathcal{L}(\mathcal{C}')$.

Proof. We shall prove that the viable behavior of any colony with bounded-life components is a context-free language.
Let $\mathcal{C} = (V, T, (S_1, F_1), (S_2, F_2), \ldots (S_n, F_n), w, bl)$ be a colony with bounded-life components. Suppose that $bl(j) = \infty$, for all $j, 1 \leq j \leq m$, and $bl(s) \in \mathcal{N}$, for all $s, m + 1 \leq s \leq n$, for some $m, 1 \leq m \leq n$.

We construct the following pushdown automaton which recognizes by final states and empty pushdown memory the language generated by $\mathcal{C}$:

$$A = (Q, T, V \cup \{Z_0\}, f, q_0, Z_0, Q \setminus \{q_e\}\}),$$

where

$$\begin{aligned} Q &= [0, bl(m+1)] \times [0, bl(m+2)] \times \ldots \times [0, bl(n)] \cup \{q_e\}, \\ q_0 &= (0, 0, \ldots, 0), \end{aligned}$$

and f is defined as follows:

$$\begin{aligned} f(q_0, \varepsilon, Z_0) &= \{(q_0, w)\}, \\ f((r_1, r_2, \ldots, r_{n-m}), \varepsilon, S_j) &= \begin{cases} \{((r_1, r_2, \ldots, r_{n-m}), x) \mid x \in F_j\}, \text{ if } j \leq m, \\ \{((r_1, r_2, \ldots, r_{n-m}), x) \mid x \in F_j\}, \text{ if } j = m + s \\ \quad \text{and } r_s + 1 \leq bl(s) \text{ for some } 1 \leq s \leq n - m, \\ \{(q_e, x) \mid x \in F_j\}, \text{ otherwise,} \end{cases} \\ f((r_1, r_2, \ldots, r_{n-m}), a, a) &= \{((r_1, r_2, \ldots, r_{n-m}), \varepsilon)\}, (r_1, r_2, \ldots, r_{n-m}) \in Q, \\ & \quad a \in T. \end{aligned}$$

The current states of the above automaton are indexed with a vector which counts the number of activations for each bounded-life component so far. If such a component has been activated more than its lifetime, then the automaton enters an error state, q_e, and the input string cannot be accepted by this computation. Thus, the input string is accepted by our automaton if and only if each component s of the index of the final state is at most $bl(s+m)$ and the pushdown memory is empty. Consequently, $L_l(\mathcal{C}) = Rec(A)$ holds. □

Given a context-free language it is natural to look for a colony with bounded-life components which generates it and has as few immortal components as possible. Thus, a natural problem concerns the possibility of limiting the life of an immortal component of a colony without modifying the viable behavior of that colony. Unfortunately, given a colony and an immortal component one cannot decide whether or not the life of this components can be limited without modifying the behaviour of the colony.

Theorem 5.2: Given a colony with bounded-life components $\mathcal{C}$ and an immortal component i of $\mathcal{C}$, one cannot decide whether or not the life of component i can be limited without modifying the viable behavior of $\mathcal{C}$.

Proof. Let $\mathcal{C}' = (V, T, (S_1, F_1), \ldots, (S_n, F_n), w)$ be an arbitrary colony with the terminal alphabet T containing at least two letters. We construct the colony with bounded-life components

$$\mathcal{C} = (V \cup \{S_{n+1}, S_{n+2}, S_{n+3}\}, T, (S_1, F_1), \ldots, (S_{n+3}, F_{n+3}), S_{n+1}, bl),$$

where

$$\begin{aligned} F_{n+1} &= \{w, S_{n+2}\}, \\ F_{n+2} &= T\{S_{n+3}\} \cup T \cup \{\varepsilon\}, \\ F_{n+3} &= T\{S_{n+2}\} \cup T, \end{aligned}$$

and

$$bl(i) = \begin{cases} 1, & \text{if } i = n+1, \\ \infty, & \text{otherwise.} \end{cases}$$

Obviously, $L_l(\mathcal{C}) = T^*$. Let us consider the immortal component $n+2$ of $\mathcal{C}$. If one bounds the life of this component by a natural number, say k, then the life of component $n+3$ can be also bounded by the same number. Let us denote by $\mathcal{C}''$ the new colony in which there is k such that $bl(n+2) = bl(n+3) = k$, i.e. the components $n+2$ and $n+3$ are not immortal anymore. Clearly, $L_l(\mathcal{C}'') = T^*$ if and only if $T^* \setminus T^{2k} \subseteq L(\mathcal{C}')$, where T^{2k} stands for the set of all strings in T^* of length smaller or equal than $2k$. Since the problem $L(\mathcal{C}') = T^*$ is known to be undecidable, it easily follows that the problem $T^* \setminus T^{2k} \subseteq L(\mathcal{C}')$ is undecidable as well, which concludes the proof. □

Now we introduce a specific descriptional complexity measure indicating the number of immortal components in a colony with bounded-life components and extend it to languages. We prove the non-triviality of this measure for context-free languages and the impossibility of computing this measure for the same class of languages.

Definition 5.3: Given a colony $\mathcal{C}$ with bounded-life components we define

$$Imm(\mathcal{C}) = \mathrm{card}(\{i \mid bl(i) = \infty\})$$

and for a context-free language L we write

$$Imm(L) = \min\{Imm(\mathcal{C}) \mid L = L_l(\mathcal{C})\}.$$

The next theorem claims that the Imm measure is not trivial.

Theorem 5.3: *For each natural number n there is a language L_n such that $Imm(L_n) \geq n$.*

Proof. For a given n we consider the context-free language

$$L_n = \{a_1^{k_1} b_1^{k_1} a_2^{k_2} b_2^{k_2} \ldots a_n^{k_n} b_n^{k_n} \mid k_1, k_2, \ldots, k_n \geq 1\}.$$

We claim that $Imm(L_n) \geq n$.
Let $\mathcal{C} = (V, \{a_1, b_1, \ldots, a_n, b_n\}, (S_1, F_1), \ldots, (S_r, F_r), x, bl)$ be a colony with bounded-life components which generates L_n. We prove the following fact

Fact: *For each nonnegative integer m there is a positive integer $p(m)$ such that*

$$card(\{i \mid K_i(a_1^{p(m)} b_1^{p(m)} \ldots a_n^{p(m)} b_n^{p(m)}, \mathcal{C}) \geq m\}) \geq n.$$

Proof of the fact. We show that for each m and some integer $p(m)$, at least n components are used at least m times in order to get $z = a_1^{p(m)} b_1^{p(m)} \ldots a_n^{p(m)} b_n^{p(m)}$. For a given positive integer m, we consider the integer $p(m) > M \cdot m \cdot r + |x|$, where

$$M = \max\{|y| \mid y \in \cup_{i=1}^{r} F_i\}.$$

Then for every a_j, $1 \leq j \leq n$, there is S_{i_j} in V such that $S_{i_j} \Longrightarrow^* a_j^{j_k} S_{i_j} x_j$ for some $j_k \leq M \cdot r$. Therefore, every derivation D of z in $\mathcal{C}$ can be written as

$$\begin{aligned} x \quad & \Longrightarrow^* a_1^{q_1} S_{i_1} x_1 \Longrightarrow^* a_1^{q_1+j_1} S_{i_1} x_2 \Longrightarrow^* a_1^{q_1+2j_1} S_{i_1} x_3 \\ & \Longrightarrow^* \ldots \Longrightarrow^* a_1^{q_1+m\cdot j_1} S_{i_1} x_{m+1} \Longrightarrow^* a_1^{p(m)} b_1^{p(m)} a_2^{q_2} S_{i_2} x_{m+2} \\ & \Longrightarrow^* a_1^{p(m)} b_1^{p(m)} a_2^{q_2+j_2} S_{i_2} x_{m+3} \Longrightarrow^* \ldots a_1^{p(m)} b_1^{p(m)} a_2^{q_2+m\cdot j_2} S_{i_2} x_{2m+2} \\ & \Longrightarrow^* \ldots \Longrightarrow^* a_1^{p(m)} b_1^{p(m)} \ldots a_n^{q_n} S_{i_n} x_{(m+1)(n-1)+2} \\ & \Longrightarrow^* a_1^{p(m)} b_1^{p(m)} \ldots a_n^{q_n+j_n} S_{i_n} x_{(m+1)(n-1)+3} \Longrightarrow^* \ldots \Longrightarrow^* \\ & a_1^{p(m)} b_1^{p(m)} \ldots a_n^{q_n+m\cdot j_n} S_{i_n} x_{(m+1)(n-1)+m+2} \Longrightarrow^* z. \end{aligned}$$

Clearly, the components $i_1, i_2, \ldots, i_n$ are mutually disjoint, otherwise parasitic strings would be obtained in $\mathcal{C}$, which concludes the proof of the fact.

Now, one can easily infer that $Imm(\mathcal{C}) \geq n$. □

Based on the previous theorem we prove that

Theorem 5.4: The measure Imm fails to be algorithmically computable.

Proof. Let $\mathcal{C}$ be a colony with bounded-life components. Assume

$$L = L(\mathcal{C})U_n^+ \cup V^* L_n,$$

where $U_n = \{a_1, b_1, a_2, b_2, \ldots, a_n, b_n\}$, $n \geq 5$, L_n is the language from the proof of the previous theorem, and $V \cap U_n = \emptyset$.

If $L(\mathcal{C}) = V^*$, then $L = V^* U_n^+$, hence $Imm(L) \leq 4$. A colony with four immortal components which generates L is

$$\mathcal{C}' = (\{S_1, S_2, S_3, S_4\} \cup V \cup U_n, V \cup U_n, (S_1, F_1), (S_2, F_2), (S_3, F_3), (S_4, F_4), S_1 S_3),$$

with

$$\begin{aligned} F_1 &= V\{S_2\} \cup V \cup \{\varepsilon\}, \\ F_2 &= V\{S_1\}, \\ F_3 &= U_n\{S_4\} \cup U_n, \\ F_4 &= U_n\{S_3\} \cup U_n. \end{aligned}$$

If $L(\mathcal{C}) \neq V^*$, then let $w \in V^* \setminus L(\mathcal{C})$ be an arbitrary word. By a similar reasoning to the proof of the previous theorem one can prove that $Imm(L) \geq Imm(\{w\}L_n)$ holds, or equivalently $Imm(L) \geq Imm(L_n)$.

Consequently, $Imm(L) \leq 4$ if and only if $L(\mathcal{C}) = V^*$, which is undecidable. Therefore one cannot compute $Imm(L)$. □

6. Colony with bounded-frequency components

In the last section we associate a sub-unit rational number with each component of a colony which limits the number of applications of that component with respect to the lenght of every derivation. Colonies with bounded-frequency components are more powerful than colonies with bounded-life components.

Definition 6.1: A colony with bounded-frequency components is a structure

$$\mathcal{C} = (V, T, (S_1, F_1), (S_2, F_2), \ldots (S_n, F_n), w, bf)$$

where

- $(V, T, (S_1, F_1), (S_2, F_2), \ldots, (S_n, F_n), w)$ is a colony, which will be denoted by $\bar{\mathcal{C}}$
- bf is a mapping from $\{1, 2, \ldots, n\}$ into the set of rationals between 0 and 1. It expresses the maximal possible frequency of using the components of $\mathcal{C}$ in the derivation.

The viable behavior of $\mathcal{C}$ is defined by

$$\begin{aligned} L_f(\mathcal{C}) &= \{x \in L(\bar{\mathcal{C}}) \mid \text{ there exists a derivation } D \text{ for } x \text{ such that} \\ & \frac{K_i(x, \mathcal{C})}{|D|} \leq bf(i), \text{ for all } 1 \leq i \leq n\}. \end{aligned}$$

Example 6.1: Let us consider the following colony with bounded-frequency components:

$$\mathcal{C} = (\{S_i | 1 \leq i \leq 12\} \cup \{a, b, c\}, \{a, b, c\}, (S_1, F_1), \ldots, (S_{12}, F_{12}), S_1S_2S_3S_4S_5S_6, bf),$$

where

$$\begin{array}{lll} F_1 = \{aS_7\} & F_2 = \{bS_8\} & F_3 = \{cS_9\} \\ F_7 = \{aS_1, a\} & F_8 = \{bS_2, b\} & F_9 = \{cS_3, c\} \\ F_4 = \{aS_{10}\} & F_5 = \{bS_{11}\} & F_6 = \{cS_{12}\} \\ F_{10} = \{aS_4, a\} & F_{11} = \{bS_5, b\} & F_{12} = \{cS_6, c\} \end{array}$$

and

$$bf(i) = \frac{1}{12}$$

for all $1 \leq i \leq 12$.

It is easy to check that $L_f(\mathcal{C}) = \{a^{2n}b^{2n}c^{2n}a^{2n}b^{2n}c^{2n} \mid n \geq 1\}$ which is not a context-free language.

Indeed, all strings generated by the above colony with bounded-frequency components are of the form $a^{2i}b^{2j}c^{2k}a^{2m}b^{2p}c^{2q}$ for some positive integers i, j, k, m, p, q. However,

$$\frac{t}{i + j + k + m + p + q} \leq \frac{1}{6}$$

for all $t \in \{i, j, k, m, p, q\}$. Let us suppose that i the the biggest integer among the others and there exists another one, say j, strictly smaller than i. Then,

$$\frac{i}{i + j + k + m + p + q} > \frac{i}{6i} = \frac{1}{6}$$

which is a contradiction. Therefore, all aforementioned integers are equal.

We recall now the definition of valence grammars from [5]. An additive valence grammar is a quintuple $G = (N, T, S, P, v)$, where (N, T, S, P) is a context-free grammar and v is a mapping from P into the set of integers Z. If $P = \{r_1, \ldots, r_n\}$, for some n, the language generated by G, denoted by $L(G)$, consists of all words x such that there is a derivation

$$S \Longrightarrow_{i_1} x_1 \Longrightarrow_{i_2} x_2 \ldots \Longrightarrow_{i_p} x_p = x$$

with $\sum_{j=1}^{p} v(r_{i_j}) = 0$.

Theorem 6.1: The class of languages generated by additive valence grammars is strictly included in the class of languages generated by colonies with bounded-frequency components.

Proof. Let L be a language generated by an additive valence grammar; by Lemma 2.1.10 from [5] we may assume that $v(p) \in \{-1, 0, 1\}$ for each rule p in P. Moreover, we may assume that for each rule $A \to x \in P$, the nonterminal A does not occur in x. Using some new nonterminals one can easily replace each rule $p \in P$ where $v(p) = 0$ with a pair of rules, without side effects on the generated language, such that for each rule r in the new set of rules, denoted also by P, $v(r) \in \{-1, 1\}$ holds. Denote by $A_1, A_2, \ldots, A_m$ the nonterminals of this last grammar generating L, $S = A_1$.

We construct the colony with bounded-frequency components

$$\mathcal{C} = (N \cup \{A_{m+1}, A_{m+2}\} \cup T, T, (A_1, F_1), (A_2, F_2), \ldots, (A_{m+2}, F_{m+2}), A_1, bf),$$

where

$$F_i = \begin{cases} \{xA_{m+1} \mid A_i \to x \in P, v(A_i \to x) = -1\} \cup \\ \{xA_{m+2} \mid A_i \to x \in P, v(A_i \to x) = 1\}, \text{ if } 1 \le i \le m \\ \{\varepsilon\}, \text{ if } i > m. \end{cases}$$

and

$$bf(i) = \begin{cases} 1, \text{ if } 1 \le i \le m \\ \frac{1}{4}, \text{ if } i > m. \end{cases}$$

Clearly, $L(\bar{\mathcal{C}})$ is exactly the language generated by the given context-free grammar, regardless the valence mapping v. It is also easy to note that, for every word x in $L_f(\mathcal{C})$, for which the last two components have been activated

i times and j times, respectively, the total length of the derivation is $2(i+j)$. Furthermore, $i = j$ must hold. Since i and j represent also the number of applications of the rules in P whose valence is -1 and 1, respectively, it follows that x can be obtained by a derivation whose total valence is 0. Hence, $L = L_f(\mathcal{C})$.

Example 2.4.1, provides a colony with bounded-frequency components which generates a language that cannot be generated by any additive valence grammar. The same idea used in Example 2.1.7 in [5] for proving that $\{a^n b^n c^n \mid n \geq 1\}^2$ cannot be generated by any additive valence grammar can be used for proving the same statement for the language from Example 1. □

The computational power of colonies with bounded-frequency components remains to be further investigated. We do not know whether or not these colonies generate recursive languages only. Clearly, the membership problem is decidable for colonies with bounded-frequency components which do not contain the empty string. But it seems to be very hard to cope with these components in the general case. Another natural problem concerns the decidability of the emptiness problem for colonies with bounded-frequency components.

References

[1] Baník, I.: Colonies with position. *Computers and Artificial Intelligence 15* (1996) 141–154

[2] Csuhaj-Varjú, E.: Colonies – a multi-agent approach to language generation. In: *Proc. ECAI'96 Workshop on Finite State Models of Language*, (A. Kornai, ed.), NJSZT, Budapest, 1996, 12–16

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

[4] Dassow, J., Kelemen, J., Păun, Gh.: On parallelism in colonies. *Cybernetics and Systems 24* (1993) 37–49

[5] Dassow, J., Păun, Gh.: *Regulated rewriting in formal language theory.* Akademie-Verlag, Berlin, 1989.

[6] Dassow, J., Păun, Gh., Rozenberg, G.: Grammar systems. In: *Handbook of Formal Languages, vol. 2* (G. Rozenberg and A. Salomaa, eds.) Springer-Verlag, Berlin, 1997, 155–214

[7] Gašo, J.: Unreliable colonies – the sequential case. *Journal of Automata, Languages and Combinatorics 5* (2000) accepted

[8] Kelemen, J., Kelemenová, A., Martin-Vide, C., Mitrana, V.: Colonies with limited activation of components. *Theoret. Comput. Sci. 244* (2000), 289–298.

[9] Kelemen, J., Kelemenová, A.: A grammar-theoretic treatment of multiagent systems. *Cybernetics and Systems 23* (1992) 621–633

[10] Kelemenová, A.: Timing in colonies. In: *Grammatical Models of Multi-Agent Systems.* (Gh. Păun and A. Salomaa, eds.) Gordon and Breach, London, 1999, 136–143

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

[12] Kelemenová, A., Kelemen, J.: From colonies to eco(grammar) systems. In: *Proc. Important results and trends in theoretical computer science*, (J. Karhumäki, H. Maurer, G. Rozenberg, eds.), LNCS 812, Springer-Verlag, Berlin, 1994, 213–231

[13] Kelemen, J., Kelemenová, A., Martín-Vide, C., Mitrana, V.: Colonies with limited activation of components. *Theoretical computer science 244/1-2* (2000) 289–298

[14] Martín-Vide, C., Păun, Gh.: PM-colonies. *Computers and Artificial Intelligence 17* (1998) 553–582

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

[16] Păun, Gh.: On the generative power of colonies. *Kybernetika 31* (1995) 83–97

[17] Păun, Gh.: A new generative device: valence grammars, *Rev. Roum. Math. Pures et Appl.*, 25, 6 (1980) 911–924.

[18] Rozenberg, G., Salomaa, A. (eds.): *The Handbook of Formal Languages.* 3 volumes, Springer-Verlag, 1996.

[19] Sosík, P.: Parallel accepting colonies and neural networks. In: *Grammatical Models of Multi-Agent Systems.* (Gh. Păun and A. Salomaa, eds.) Gordon and Breach, London, 1999, 144–156

[20] Sosík, P., Štýbnar, L.: Grammatical inference of colonies. In: *New Trends in Formal Languages.* (Gh. Păun and A. Salomaa, eds.) Springer-Verlag, Berlin, 1997, 236–246

[21] Štýbnar, J.: Time dependent behaviour of components in colonies. Diploma thesis. Silesian University, Opava, 1999