

Computing, Interaction and Multi-Agent Systems: A Formal Language Approach

Gemma Bel-Enguix and M. Dolores Jiménez-López

Research Group on Mathematical Linguistics (GRLMC)

Rovira i Virgili University

Pl. Imperial Tàrraco, 1, 43005 Tarragona, Spain

`gemma.bel@urv.cat`

`mariadolores.jimenez@urv.cat`

Abstract. Starting from the metaphor of computation as *interaction*, we introduce a multi-agent system framework based on a non-classical formal language model and inspired in one of the most common forms of interaction: *dialogue*.

1 Introduction

According to [5], computing has enjoyed several different metaphors for the notion of computation. Until the mid-1960s, most people thought of computation as *calculation*, or operations undertaken on numbers. From the 1960s, computation was re-conceptualised more generally as *information processing*, or operations on text, audio or video data. With the growth of the Internet over the last fifteen years, a new metaphor for computation has appeared: computation as *interaction*.

In order to exploit this new metaphor of computing, many people deal with agent technologies. Agent-based systems are one of the most important areas of research and development that have emerged in information technology in the 1990s. Roughly speaking, an agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, multi-agent domains.

The concept of agent can be found in a range of disciplines as, for example, computer networks, software engineering, artificial intelligence, human-computer interaction, distributed and concurrent systems, mobile systems, telematics, information retrieval, etc. In general, multi-agent systems offer strong models for representing complex and dynamic real-world environment.

According to [5], agent technologies can be grouped into three categories, according to the scale at which they apply:

1. *Organization-Level*: technologies and techniques related to agent societies as a whole. Here, issues of organizational structure, trust, norms and obligations and self-organization in open agent societies are paramount.
2. *Interaction-Level*: technologies and techniques that concern the communications between agents –for example, technologies related to communication languages, interaction protocols and resource allocation mechanisms.

3. *Agent-Level*: technologies and techniques concerned only with individual agents –for example, procedures for agent reasoning and learning.

In this paper we introduce a multi-agent system model based on grammar systems that can be placed at the *interaction-level*. In general, problems solved by technologies on this level have been studied in other disciplines such as economics, political science, philosophy and linguistics. In fact, the framework we present in this paper has been inspired on linguistics, specifically on the functioning of natural language dialogue. Natural Language can be seen not just a system of communication but also as a coordination system. The capacity of natural language to play these two roles has been used in order to develop an interactive model that integrate coordination and communication in a multi-agent system.

Throughout the paper, we assume that the reader is familiar with the basics of formal language theory, for more information see [8] and [7].

2 Conversational Grammar Systems

Grammar systems theory is a consolidated and active branch in the field of formal languages [1] that provides syntactic models for describing multi-agent systems at the symbolic level, using tools from formal grammars and languages. The attempt of the ‘parents’ of the theory was, as themselves state in [1], ‘to demonstrate a particular possibility of studying complex systems in a purely syntactic level’ or, what is the same, to propose a *grammatical* framework for multi-agent systems. Grammar systems theory has been widely investigated and nowadays constitutes a well-developed formal theory that presents several advantages with respect to classical models. However, being a branch of formal languages, researchers in the field of grammar systems have concentrated mainly on theoretical aspects. Roughly speaking, a grammar system is a *set* of grammars working together, according to a specified protocol, to generate a language. Notice that while in classical formal language theory *one* grammar (or automata) works individually to generate (or recognize) *one* language; here, instead, we have *several* grammars working together in order to produce *one* language.

While grammar systems are related to AI, a subfield of the theory, –the so-called eco-grammar systems– is closely related to Artificial Life. Eco-grammar systems provide a syntactical framework for eco-systems, this is, for communities of evolving agents and their interrelated environment. Briefly, an eco-grammar system is defined as a multi-agent system where different components, apart from interacting among themselves, interact with a special component called ‘environment’ [6].

Here we introduce a new model: *Conversational Grammar Systems* (CGS). CGS are multi-agent systems based on grammar systems, specifically in the so-called eco-grammar systems. Conversational grammar system offer a framework with a high degree of flexibility, what means that they are able to accept new concepts and modify rules, protocols and settings during the computation.

Evolution and action are involved in a consistent way in environment/contexts, while interaction of agents with the medium is constant. Moreover, conversational grammar systems present the following advantages to exploit the idea of computation as interaction:

- generation process is highly *modularised* by a distributed system of contributing agents;
- it is *contextualized*, linguistic agents re-define their capabilities according to context conditions given by mappings;
- and *emergent*, it emerges from current competence of the collection of active agents.

In what follows we introduce the formal definition of our model.

3 Elementary Components

In conversational grammar systems we distinguish two types of components: *agents* and *environment*. These elementary components are defined in the following way:

Definition 1 A *Conversational Grammar System (CGS)* of degree n , $n \geq 2$, is an $(n + 1)$ -tuple:

$$\Sigma = (E, A_1, \dots, A_n),$$

where:

- $E = (V_E, P_E)$,
 - V_E is an alphabet;
 - P_E is a finite set of rewriting rules over V_E .
- $A_i = (V_i, P_i, R_i, \varphi_i, \psi_i, \pi_i, \rho_i)$, $1 \leq i \leq n$,
 - V_i is an alphabet;
 - P_i is a finite set of rewriting rules over V_i ;
 - R_i is a finite set of rewriting rules over V_E ;
 - $\varphi_i: V_E^* \rightarrow 2^{P_i}$;
 - $\psi_i: V_E^* \times V_i^+ \rightarrow 2^{R_i}$;
 - π_i is the start condition;
 - ρ_i is the stop condition;
 - π_i and ρ_i are predicates on V_E^* . We can define the following special types of predicates. We say that predicate σ on V_E^* is of:
 - * Type (a) iff $\sigma(w) = \text{true}$ for all $w \in V_E^*$;
 - * Type (rc) iff there are two subsets R and Q of V_E and $\sigma(w) = \text{true}$ iff w contains all letters of R and w contains no letter of Q ;
 - * Type (K) iff there are two words x and x' over V_E and $\sigma(w) = \text{true}$ iff x is a subword of w and x' is not a subword of w ;

- * Type (K') iff there are two finite subsets R and Q of V_E^* and $\sigma(w) = \text{true}$ iff all words of R are subwords of w and no word of Q is a subword of w ;
- * Type (C) iff there is a regular set R over V_E and $\sigma(w) = \text{true}$ iff $w \in R$.

The items of the above definition have been interpreted as follows: a) E represents the environment described at any moment of time by a string w_E , over alphabet V_E , called the *state of the environment*. The state of the environment is changed both by its own evolution rules P_E and by the actions of the agents of the system, A_i , $1 \leq i \leq n$. b) A_i , $1 \leq i \leq n$, represents an agent. It is identified at any moment by a string of symbols w_i , over alphabet V_i , which represents its current state. This state can be changed by applying evolution rules from P_i , which are selected according to mapping φ_i and depend on the state of the environment. A_i can modify the state of the environment by applying some of its action rules from R_i , which are selected by mapping ψ_i and depend both on the state of the environment and on the state of the agent itself. Start/Stop conditions of A_i are determined by π_i and ρ_i , respectively. A_i starts/stops its actions if context matches π_i and ρ_i . Start/stop conditions of A_i can be of different types: (a) states that an agent can start/stop at any moment. (rc) means that it can start/stop only if some letters are present/absent in the current sentential form. And (K) , (K') and (C) denote such cases where global context conditions have to be satisfied by the current sentential form.

4 Elementary Configurations

In CGS, we define an elementary configuration as a *state* in which the system can be at a given time.

Definition 2 A state of a CGS $\Sigma = (E, A_1, \dots, A_n)$, $n \geq 2$, is an $n + 1$ -tuple:

$$\sigma = (w_E; w_1, \dots, w_n),$$

where $w_E \in V_E^*$ is the state of the environment, and $w_i \in V_i^*$, $1 \leq i \leq n$, is the state of agent A_i .

5 Behavior of Agents

The behavior of agents in CGS consists in the application of action rules to the state of the environment. We describe this behavior as a sequence of *context-change-actions* allowed by the current environment and performed by two or more agents. An *action* is defined as the application of a rule on the *environmental string*. This rule is applied to the state of the environment by an *active agent*, and it is not any rule, but a rule selected by $\psi_i(w_E, w_i)$, that is, a rule (an action) allowed by the current context and by the state of the agent itself. We define an *active agent* in relation to the allowable actions it has at a given moment. Formally:

Definition 3 By an action of an active agent A_i in state $\sigma = (w_E; w_1, w_2, \dots, w_n)$ we mean a direct derivation step performed on the environmental state w_E by the current action rule set $\psi_i(w_E, w_i)$ of A_i .

Definition 4 An agent A_i is said to be active in state $\sigma = (w_E; w_1, w_2, \dots, w_n)$ if the set of its current action rules, that is, $\psi_i(w_E, w_i)$, is a nonempty set.

6 Environmental Dynamics

Since interaction in CGS is understood in terms of *context changes*, we have to define how the environment passes from one state to another as a result of agents' actions:

Definition 5 Let $\sigma = (w_E; w_1, \dots, w_n)$ and $\sigma' = (w'_E; w'_1, \dots, w'_n)$ be two states of a CGS $\Sigma = (E, A_1, \dots, A_n)$. We say that σ' arises from σ by a simultaneous action of active agents $A_{i_1}, \dots, A_{i_r}$, where $\{i_1, \dots, i_r\} \subseteq \{1, \dots, n\}$, $i_j \neq i_k$, for $j \neq k$, $1 \leq j, k \leq r$, onto the state of the environment w_E , denoted by $\sigma \xrightarrow{a}_{\Sigma} \sigma'$, iff:

- $w_E = x_1 x_2 \dots x_r$ and $w'_E = y_1 y_2 \dots y_r$, where x_j directly derives y_j by using current rule set $\psi_i(w_E, w_{i_j})$ of agent A_{i_j} , $1 \leq j \leq r$;
- there is a derivation:

$$w_E = w_0 \xrightarrow{a}_{A_{i_1}}^* w_1 \xrightarrow{a}_{A_{i_2}}^* w_2 \xrightarrow{a}_{A_{i_3}}^* \dots \xrightarrow{a}_{A_{i_r}}^* w_r = w'_E$$

such that, for $1 \leq j \leq r$, $\pi_{i_j}(w_{j-1}) = \text{true}$ and $\rho_{i_j}(w_j) = \text{true}$. And for $f \in \{t, \leq k, \geq k\}$ the derivation is:

$$w_E = w_0 \xrightarrow{a}_{A_{i_1}}^f w_1 \xrightarrow{a}_{A_{i_2}}^f w_2 \xrightarrow{a}_{A_{i_3}}^f \dots \xrightarrow{a}_{A_{i_r}}^f w_r = w'_E$$

such that, for $1 \leq j \leq r$, $\pi_{i_j}(w_{j-1}) = \text{true}^1$, and

- $w'_i = w_i$, $1 \leq i \leq n$.

However, in the course of the computation, agents' states are also modified and the environmental string is subject to changes due to reasons different from agents' actions. So, in order to complete our formalization, we add the following definition:

Definition 6 Let $\sigma = (w_E; w_1, \dots, w_n)$ and $\sigma' = (w'_E; w'_1, \dots, w'_n)$ be two states of a CGS $\Sigma = (E, A_1, \dots, A_n)$. We say that σ' arises from σ by an evolution step, denoted by $\sigma \xrightarrow{e}_{\Sigma} \sigma'$, iff the following conditions hold:

- w'_E can be directly derived from w_E by applying rewriting rule set P_E ;

¹ In this latter case the stop condition $\rho_i(w_j) = \text{true}$ is replaced by the stop condition given the f -mode.

- w'_i can be directly derived from w_i by applying rewriting rule set $\varphi_i(w_E)$, $1 \leq i \leq n$.

In CGS, computation implies that both the *state of the environment* and *state of agents* change. Such changes take place thanks to two different types of processes: *action steps* and *evolution steps*. By means of the former, active agents perform actions on the environmental string modifying its state; the latter imply the reaction of context and agents which, according to the changes produced by agents' actions, modify their states. So, action steps and evolution steps alternate in the course of the computation. At the end, what we have is a *sequence of states* reachable from the initial state by performing, alternatively, action and evolution derivation steps:

Definition 7 Let $\Sigma = (E, A_1, \dots, A_n)$ be a CGS and let σ_0 be a state of Σ . By a *state sequence* (a *derivation*) starting from an initial state σ_0 of Σ we mean a sequence of states $\{\sigma_i\}_{i=0}^\infty$, where:

- $\sigma_i \xrightarrow{a} \Sigma \sigma_{i+1}$, for $i = 2j$, $j \geq 0$; and
- $\sigma_i \xrightarrow{e} \Sigma \sigma_{i+1}$, for $i = 2j + 1$, $j \geq 0$.

Definition 8 For a given CGS Σ and an initial state σ_0 of Σ , we denote the set of state sequences of Σ starting from σ_0 by $Seq(\Sigma, \sigma_0)$.

The set of environmental state sequences is:

$$Seq_E(\Sigma, \sigma_0) = \{\{w_{Ei}\}_{i=1}^\infty \mid \{\sigma_i\}_{i=0}^\infty \in Seq(\Sigma, \sigma_0), \sigma_i = (w_{Ei}; w_{1i}, \dots, w_{ni})\}.$$

The set of state sequences of the j -th agent is defined by:

$$Seq_j(\Sigma, \sigma_0) = \{\{w_{ji}\}_{i=1}^\infty \mid \{\sigma_i\}_{i=0}^\infty \in Seq(\Sigma, \sigma_0), \sigma_i = (w_{Ei}; w_{1i}, \dots, w_{ji}, \dots, w_{ni})\}.$$

$Seq(\Sigma, \sigma_0)$ describes the behavior of the system, this is, the possible state sequences, directly following each other, starting from the initial state. $Seq_E(\Sigma, \sigma_0)$ and $Seq_j(\Sigma, \sigma_0)$ are the corresponding sets of sequences of the states of the environment and of the states of j -th agent, respectively.

Now, we associate certain languages with an initial configuration:

Definition 9 For a given CGS Σ and an initial state σ_0 of Σ , the *language of the environment* is:

$$L_E(\Sigma, \sigma_0) = \{w_E \in V_E^* \mid \{\sigma_i\}_{i=0}^\infty \in Seq(\Sigma, \sigma_0), \sigma_i = (w_E; w_1, \dots, w_n)\}.$$

and the *language of j -th agent* is:

$$L_j(\Sigma, \sigma_0) = \{w_j \in V_A^* \mid \{\sigma_i\}_{i=0}^\infty \in Seq(\Sigma, \sigma_0), \sigma_i = (w_E; w_1, \dots, w_j, \dots, w_n)\}.$$

for $j = 1, 2, \dots, n$.

$L_E(\Sigma, \sigma_0)$ and $L_j(\Sigma, \sigma_0)$ correspond to those states of the environment and to those states of the j -th agent, respectively, that are reachable from the initial configuration of the system.

7 Interaction Protocol

Coordination is defined in many ways but in its simplest form it refers to ensuring that the actions of independent actors (agents) in an environment are coherent in some way. The challenge therefore is to identify mechanisms that allow agents to coordinate their actions. Research to date has identified a huge range of different types of coordination and cooperation mechanisms, ranging from emergent cooperation, coordination protocols to distributed planning. In CGS, we define different *modes of derivation* that can be seen as the interaction protocol of our multi-agent system:

Definition 10 Let $\Sigma = (E, A_1, \dots, A_n)$ be a CGS. And let $w_E = x_1x_2\dots x_r$ and $w'_E = y_1y_2\dots y_r$ be two states of the environment. Let us consider that w'_E directly derives from w_E by action of active agent A_i , $1 \leq i \leq n$, as shown in Definition 5. We write that:

$$\begin{aligned} w_E &\xrightarrow{a}_{A_i}^{\leq k} w'_E \text{ iff } w_E \xrightarrow{a}_{A_i}^{\leq k'} w'_E, \text{ for some } k' \leq k; \\ w_E &\xrightarrow{a}_{A_i}^{\geq k} w'_E \text{ iff } w_E \xrightarrow{a}_{A_i}^{\leq k'} w'_E, \text{ for some } k' \geq k; \\ w_E &\xrightarrow{a}_{A_i}^* w'_E \text{ iff } w_E \xrightarrow{a}_{A_i}^k w'_E, \text{ for some } k; \\ w_E &\xrightarrow{a}_{A_i}^t w'_E \text{ iff } w_E \xrightarrow{a}_{A_i}^* w'_E \text{ and there is no } z \neq y \text{ with } y \xrightarrow{a}_{A_i}^* z. \end{aligned}$$

In words, $\leq k$ -derivation mode represents a time limitation where A_i can perform at most k successive actions on the environmental string. $\geq k$ -derivation mode refers to the situation in which A_i has to perform at least k actions whenever it participates in the derivation process. With $*$ -mode, we refer to such situations in which agent A_i performs as many actions as it wants to. And finally, t -derivation mode represents such cases in which A_i has to act on the environmental string as long as it can.

One way of getting transitions with no gap and no overlap in CGS is to endow agents with an *internal control* that contains start/stop conditions that allow agents to recognize places where they can start their activity, as well as places where they should stop their actions and give others the chance to act. This is, start/stop conditions help agents to recognize *transition relevance places*, i.e. places where speaker change occurs. Start/stop conditions have been formally defined in Definition 1.

Figure 1 gives a graphic idea of the multi-agent system architecture we have introduced.

8 Example

The following simple example illustrates how CGS work.

Example 1 Consider the following CGS: $\Sigma = (E, A_1, A_2)$, where:

$$- E = (V_E, P_E),$$

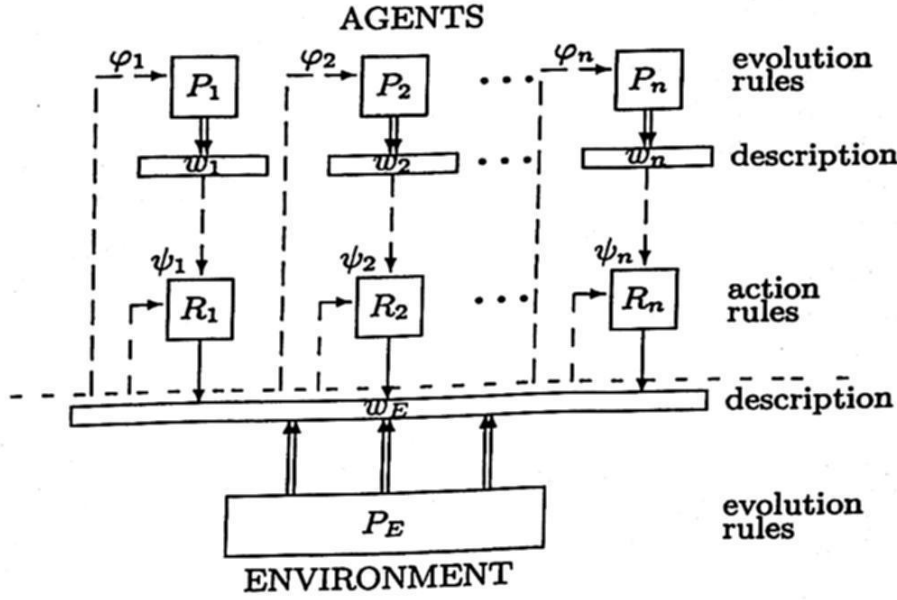


Fig. 1. Conversational Grammar Systems.

- $V_E = \{a, x, y\}$;
- $P_E = \{a \rightarrow b^2, b \rightarrow a^2, x \rightarrow x, y \rightarrow y\}$.
- $A_1 = (V_1, P_1, R_1, \varphi_1, \psi_1, \pi_1, \rho_1)$ with:
 - $V_1 = \{c\}$;
 - $P_1 = \{c \rightarrow c\}$; $R_1 = \{a \rightarrow x\}$;
 - $\varphi_1(w) = P_1$ for every $w \in V_E^*$;
 - $\psi_1(w; u) = R_1$ for $w \in \{a, x, y\}^*$ and $u = c$, otherwise $\psi_1(w; u) = \emptyset$;
 - $\pi_1 = \text{true}$ for all $w \in V_E^*$; $\rho_1 = \text{true}$ for all $w \in V_E^*$.
- $A_2 = (V_2, P_2, R_2, \varphi_2, \psi_2, \pi_2, \rho_2)$ with:
 - $V_2 = \{d\}$;
 - $P_2 = \{d \rightarrow d\}$; $R_2 = \{b \rightarrow y\}$;
 - $\varphi_2(w) = P_2$ for every $w \in V_E^*$;
 - $\psi_2(w; v) = R_2$ for $w \in \{b, x, y\}^*$ and $v = d$, otherwise $\psi_2(w; v) = \emptyset$;
 - $\pi_2 = \text{true}$ for all $w \in V_E^*$; $\rho_2 = \text{true}$ for all $w \in V_E^*$.

P_E , P_1 and P_2 contain rules of an OL system applied in a parallel way. Rules in R_1 and R_2 are pure context-free productions applied sequentially. Let us suppose that the system is working in the arbitrary mode *. And let us take $\sigma_0 = (a^3; c, d)$ as the initial state of Σ . Then, a possible derivation in Σ is the following one:

$$\begin{aligned}
 (a^3; c, d) &\xRightarrow{a}_{\Sigma}^* (a^2x; c, d) \xRightarrow{e}_{\Sigma}^* (b^4x; c, d) \xRightarrow{a}_{\Sigma}^* (yb^3x; c, d) \xRightarrow{e}_{\Sigma}^* \\
 &\quad (ya^6x; c, d) \xRightarrow{a}_{\Sigma}^* (ya^2xa^3x; c, d) \xRightarrow{e}_{\Sigma}^* \dots
 \end{aligned}$$

Notice, that we alternate action and evolution steps. At every action step one of the agents rewrites one symbol of the environmental state, while in evolution steps both environmental and agents' states are rewritten according to OL rules.

9 Final Remarks

Grammar Systems provide a well-defined *theoretical* formal model for multi-agent systems with interesting and well-known formal results. Taking into account this formal language model and considering the functioning of interaction in natural language we have defined a new model called *conversational grammar systems*. The core ideas of the model we have introduced here are the following ones:

- *Multi-agent System*. Conversational grammar systems can be seen as a multi-agent system for computation. CGSs define systems of distributed components in which components can be viewed as autonomous problem solvers that must collaborate in order to perform complex tasks.
- *Distribution*. We have a multi-agent system of *autonomous* agents where the functionality of an agent is viewed as an emergent property of its intensive interaction with its dynamic environment. Each autonomous agent may accomplish its own task, or cooperate with other agents, to perform its own individual task or a global social one.
- *Dynamic, emergent*. Conversational grammar systems offers a dynamic and emergent model for interaction. Conversational grammar systems have been defined as a set of agents developing their activity on a common shared environment. And we have emphasized very much the fact that actions performed by agents are determined both by the state of context and by the state of the agent itself. We have not postulated any external control to fix the sequence of actions to be performed during derivation process. Which actions must be performed at any moment is a matter solved *locally* and *opportunisticly*, by taking into account which the state of context at that precisely moment is.

We claim that CGS provides a powerful framework for formalizing any kind of *interaction*, both among agents and among agents and the environment. A topic where context and interaction among agents is essential is the field of dialogue modelling and its applications to the design of effective and user-friendly computer dialogue systems where we think our model can be directly applied.

Our model has been defined as a multi-agent system. Agent technology is one of the fastest growing areas of information technology. People agree on the fact that the apparatus of agent technology provides a powerful a useful set of structures and processes for designing and building complex software applications. The metaphor of autonomous problem solving entities cooperating and coordinating to achieve their objectives is a natural way of conceptualizing many problems. Being conversational grammar systems an agent-based model, they share all those advantageous features. Therefore, we think that our model may contribute to the field of agent technologies by offering a highly formalized framework that could be applied to many different issues.

References

1. Csuhaj-Varjú, E., Dassow, J., Kelemen, J. & Păun, Gh. (1994), *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*, Gordon and Breach, London.
2. Helander, M.G., Landauer, T.K. & Prabhu, P.V. (1997), *Handbook of Human-Computer Interaction*, Elsevier, Amsterdam.
3. Jennings, N.R. & Wooldridge, M.J. (1998), *Agent Technology. Foundations, Applications and Markets*, Springer, Berlin.
4. Kraus, S. (1997), Negotiation and Cooperation in Multi-Agent Environments, *Artificial Intelligence*, 94, 79-97.
5. Luck, M., McBurney, P., Shehory, O. & Willmott, S. (eds.) (2005), *Agent technology: Computing as Interaction. A Roadmap for Agent Based Computing*, University of Southampton (AgentLink III).
6. Păun, Gh. (ed.) (1995), *Artificial Life: Grammatical Models*, Black Sea University Press, Bucharest.
7. Rozenberg, G. & Salomaa, A. (1997), *Handbook of Formal Languages*. Springer, Berlin.
8. Salomaa, A. (1973), *Formal Languages*, Academic Press, New York.
9. Tatai, G. & Gulyás, L. (eds.) (1999), *Agents Everywhere*, Springer, Budapest.