# Learning, Agents and Formal Languages
## *State of the Art*

Leonor Becerra-Bonache[1] and M. Dolores Jiménez-López[2]

[1]*Laboratoire Hubert Curien, Université Jean Monnet, Rue du Professeur Benoit Lauras 18, 42000 Saint Etienne, France*

[2]*Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Av. Catalunya 35, 43002 Tarragona, Spain*

Keywords: Machine Learning, Agent Technology, Formal Languages.

Abstract: The paper presents the state of the art of machine learning, agent technologies and formal languages, not considering them as isolated research areas, but focusing on the relationship among them. First, we consider the relationship between *learning* and *agents*. Second, the relationship between *machine learning* and *formal languages*. And third, the relationship between *agents* and *formal languages*. Finally, we point to some promising directions on the intersection among these three areas.



# 1 INTRODUCTION

This paper focuses on the common space delimited by three main areas: *machine learning*, *agent technology* and *formal language theory*.

Understanding human learning well enough to reproduce aspects of that learning capability in a computer system is a worthy scientific goal that have been considered by the research on *machine learning*, a field of Artificial Intelligence that aims to develop techniques that allow computers to learn. As Nilsson says,"a machine learns whenever it changes its structure, program or data (based on its inputs or in response to external information) in such a manner that its expected future performance improves" (Nilsson, 1998). Machine learning techniques have been successfully applied to different domains, such us bioinformatics (e.g., gene finding), natural language processing (e.g., machine translation), speech and image recognition, robotics, etc.

*Agent technology* is one of the most important areas of research and development that have emerged in information technology in the 1990s. It can be defined as a Distributed Artificial Intelligence approach to implement autonomous entities driven by beliefs, goals, capabilities, plans and agency properties. Roughly speaking, an agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, multi-agent domains. The metaphor of autonomous problem solving entities cooperating and coordinating to achieve their objectives is a natural way of conceptualizing many problems. In fact, the multi-agent system literature spans a wide range of fields including robotics, mathematics, linguistics, psychology, and sociology, as well as computer science.

*Formal languages* originated from mathematics and linguistics as a theory that provides mathematical tools for the description of linguistic phenomena. The main goal of formal language theory is the syntactic finite specification of infinite languages. The theory was born in the middle of the 20th century as a tool for modeling and investigating the syntax of natural languages. However, very soon it developed as a new research field, separated from linguistics, with specific problems, techniques and results and, since then, it has had an important role in the field of computer science, in fact it is considered as the stem of theoretical computer science.

Our goal here is to provide a state of the art of these three areas, but not considering them as isolated research topics, but by focusing in the relationship among them (see Figure 1). The organization of the paper is as follows.

In section 2, we consider the relationship between *learning* and *agents*. The intersection of multi-agent systems and machine learning techniques have given rise to two different research areas (Jedrzejowicz, 2011): 1) *learning in multi-agent systems* where machine learning solutions are applied to support agent technology and 2) *agent-based machine learning techniques* where agent technology is used in the field machine learning with the interest on applying agent-based solutions to learning.


Becerra Bonache L. and Jiménez-López M..
Learning, Agents and Formal Languages - State of the Art.
DOI: 10.5220/0004359704700478
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (LAFLang-2013), pages 470-478
ISBN: 978-989-8565-38-9

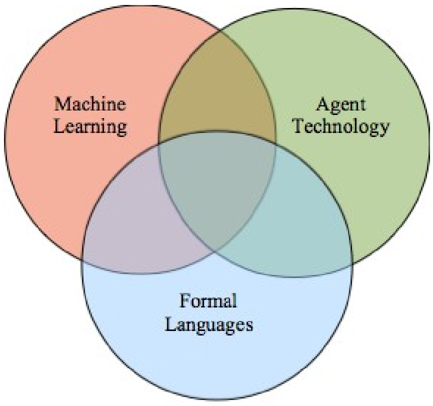


Figure 1: Intersection among *machine learning*, *agent technology* and *formal language theory*.

In section 3, the relationship between *learning* and *formal languages* is taken into account. The theory of formal language theory is central to the field of machine learning, since the area of *grammatical inference* –a subfield of machine learning– deals with the process of learning formal grammars and languages from a set of data.

In section 4, the relationship between *agents* and *formal languages* is considered. While in classic formal language theory, grammars and automata modeled classic computing devices where the computation was accomplished by one central agent, new models in formal languages take into account distributed computation. The idea of several devices collaborating for achieving a common goal was formalized in many subfields of formal language theory giving rise to the so-called *agent-based models* of formal languages.

Finally, section 5 concludes the paper by suggesting potential and promising directions of future research on the intersection among learning, agents and formal languages.

## 2 LEARNING AND AGENTS

The intersection of agent technology and machine learning constitutes a research area whose importance is nowadays broadly acknowledged in artificial intelligence: *learning in multi-agent systems*. This new area has emerged as a topic of research in the late 1980s and since then has attracted increasing attention in both the multi-agent systems community and the machine learning area. However, until the late 80s, multi-agent learning had been widely ignored by both researchers in distributed artificial intelligence and in machine learning. This situation was due to two facts: 1) work in distributed artificial intelligence mainly concentrated on developing multi-agent systems whose organization and functioning were fixed and, 2) research in machine learning mainly concentrated on learning techniques and methods for single-agent settings (Weiss and Dillenbourg, 1998).

Nowadays, it is commonly agreed by distributed artificial intelligence and machine learning communities that multi-agent learning –this is, learning that requires the interaction among several intelligent agents (Huhns and Weiss, 1998)-deserves particular attention. Two important reasons for the interest in studying learning in multi-agent systems have been stressed (Weiss, 1993):

1. The need for learning techniques and methods in the area of multi-agent systems in order to equip multi-agent systems with learning abilities to allow agents to automatically improve their behavior.
2. The need in the area of machine learning area of considering not only single-agent learning but also multi-agent learning in order to improve the understanding of the learning processes in natural multi-agent systems (like human groups or societies).

The area of multi-agent learning shows how developments in the fields of machine learning and agent technologies have become complementary. In this intersection, researchers from both fields have opportunities to profit from solutions proposed by each other. In fact we can distinguish two directions in this intersection (Jedrzejowicz, 2011):

1. *Learning in Multi-Agent Systems* (MAS), this is, using machine learning techniques in agent technology.
2. *Agent-Based Machine Learning*, this is, using agent technology in the field of machine learning.

### 2.1 Learning in Multi-agent Systems

Learning is increasingly being seen as a key ability of agents and, therefore, several agent-based frameworks that utilize machine learning for intelligent decision support have been reported. Theoretical developments in the field of learning agents focus mostly on methodologies and requirements for constructing multi-agent systems with learning capabilities.

Many terms can be found in the literature that refer to learning in multi-agent systems (Sen and Weiss, 1999): mutual learning, cooperative learning, collaborative learning, co-learning, team learning, social learning, shared learning, pluralistic learning, and organizational learning are just some examples.

In the area of multi-agent learning –the application of machine learning to problems involving multiple agents (Panait and Luke, 2005)–, two principal forms of learning can be distinguished (Sen and Weiss, 1999; Weiss, 1993):

1. *Centralized or isolated learning* where the learning process is executed by one single agent and does not require any interaction with other agents.
2. *Decentralized, distributed, collective or interactive learning* where several agents are engaged in the same learning process and the learning is done by the agents as a group.

There are three main methods/approaches to learning in multi-agent systems which are distinguished by taking into account the kind of feedback provided to the learner (Panait and Luke, 2005; Weiss, 1993).

1. *Supervised learning*, where the correct output is provided. This means that the environment or an agent providing feedback acts as a "teacher".
2. *Reinforcement learning*, where an assessment of the learner's output is provided. This means that the environment or an agent providing feedback acts as a "critic".
3. *Unsupervised learning*, where no explicit feedback is provided at all. This means that the environment or an agent providing feedback acts as an "observer".

Space limitation prevents us of going deeper in the above models. For more information the reader can see (Panait and Luke, 2005; Weiss, 1993; Weiss and Dillenbourg, 1998; Weiss, 1998; Stone and Veloso, 2000; Shoham et al., 2007; Sen and Weiss, 1999). Our goal in this section has been just to stress the fact that several dimensions of multi-agent interaction can be subject to learning –when to interact, with whom to interact, how to interact, and what exactly the content of the interaction should be (Huhns and Weiss, 1998)–, and machine learning can be seen as a primer supplier of learning capabilities for agent and multi-agent systems.

## 2.2 Agent-based Machine Learning

In the intersection between multi-agent systems and machine learning we find the so-called *agent-based machine learning techniques* where agent technology is applied to solve machine learning problems. According to Jedrzejowicz (Jedrzejowicz, 2011), there are several ways in which the research of machine learning can profit from the application of agent technology:

- First of all, there are machine learning techniques where parallelization can speed-up learning, therefore, in these cases using a set of agents may increase the efficiency of learning.
- Secondly, there are machine learning techniques that rely on the collective computational intelligence paradigm, where a synergetic effect is expected from combining efforts of various agents.
- Thirdly, in the so-called distributed machine learning problems, a set of agents working in distributed sites can be used to produce some local level solutions independently and in parallel.

Taking into account those advantages, several models have been proposed that apply agent-based solutions to machine learning problems:

- Models of collective or collaborative learning.
- Learning classifier systems that use agents representing set of rules as a solution to machine learning problem.
- Ensemble techniques.
- Distributed learning models.

According to (Jedrzejowicz, 2011), agent technology has brought to machine learning several capabilities including parallel computation, scalability and interoperability. In general, agent based solutions can be used to develop more flexible machine learning tools. For the state of the art of agent-based machine learning see (Jedrzejowicz, 2011).

# 3 LEARNING AND FORMAL LANGUAGES

The intersection between machine learning and formal languages constitutes a well-established research area known as *grammatical inference*. As A. Clark says "Grammatical inference is the study of machine learning of formal languages" (Clark, 2004). This new area was born in the 1960s and since then has attracted the attention of researchers working on different fields, including machine learning, formal languages, automata theory, computational linguistics, information theory, pattern recognition, and many others.

E.M. Gold (Gold, 1967) originated the study of grammatical inference and gave the initial theoretical foundations of this field. Motivated by the problem of children's language acquisition, E.M. Gold aimed "to construct a precise model for the intuitive notion able to speak a language in order to be able to investigate theoretically how it can be achieved artificially"

(Gold, 1967). After Gold's work, there has been developed a considerable amount of work to established a grammatical inference theory, to find efficient methods for inferring formal grammars, and to apply those methods to practical domains, such as bioinformatics or natural language processing.

As H. Fernau and C. de la Higuera pointed out (Fernau and de la Higuera, 2004), there is a number of good reasons for formal language specialists to be interested in the field of grammatical inference, among others:

- Grammatical inference deals with formalisms describing formal languages, such us formal grammars, automata, etc.
- Grammatical inference uses formal language methodologies for constructing learning algorithms and for reasoning about them.
- Grammatical inference tries to give mathematical descriptions of the classes of languages that can be learned by a concrete learning algorithm.

Most of grammatical inference research has been focused on learning *regular* and *context-free* languages. Although these are the smallest classes of the Chomsky hierarchy, it has been proved that even to learn these classes is already too hard under certain learning paradigms. Next, we will review the main formal models proposed in this field and some of the main learnability results obtained.

## 3.1 Learning Paradigms

Broadly speaking, in a grammatical inference problem, we have a *teacher* that provides data to the learner (or learning algorithm), and a *learner* that must identify the underlying language from this data. Depending on the kind of data given to the learner, how this data is provided to it and the criteria used to say that a learner has successfully learnt the language, we can distinguish three main learning paradigms:

- Identification in the limit, proposed by Gold (Gold, 1967).
- Query learning, proposed by Angluin (Angluin, 1987).
- Probably Approximately Correct learning (PAC), proposed by Valiant (Valiant, 1984).

Imagine an adult and a child learning his native language. The adult uses his grammar, $G$, to construct sentences of his language, $L$. The child receives sentences and, after some time, he is able to use grammar $G$ to construct sentences of $L$. From a mathematical point of view, the child is described by a learning algorithm, which takes a list of sentences as input and generates a language as output. Based on these ideas, Gold introduced a new formal model known as *identification in the limit* (Gold, 1967), with the ultimate goal of explaining the process of children's language acquisition. In this model, examples of the unknown language are presented to the learner, and the learner has to produce a hypothesis of this language. Its hypothesis is updated after receiving each example; if the new examples received are not consistent with the current hypothesis, it changes its hypothesis. However, at some point, always, the learner will found the correct hypothesis and will not change from it. Therefore, according to Gold's model, the learner identifies the target language in the *limit* if after a finite number of examples, the learner makes a correct hypothesis and those not change it from there on.

There are two traditional settings within Gold's model: a) learning from text, where only examples of the target language are given to the learner (i.e., only *positive data*); b) learning from informant, where examples that belong and do not belong to the target language are provided to the learner (i.e., *positive and negative* information).

It is desirable that learning can be achieved from only positive data, since in the most part of applications the available data is positive. However, one of the main Gold's results is that *superfinite classes* of languages (i.e., classes of languages that contains all finite languages and at least one infinte language) *are not identifiable in the limit* from *positive data*. This implies that even the class of regular languages is not identifiable in the limit from positive data. The intuitive idea is that, if the target language is a finite language contained in an infinite language, and the learner infers that the target language is the infinite language, it will not have any evidence to refute its hypothesis and it will never converge to the correct language. Due to these results, learning from only positive data is considered a hard task. However, learnability results have been obtained by studying subclasses of the languages to be learned, providing additional information to the learner, etc. For more details, see (de la Higuera, 2010).

In Gold's model, the learner *passively* receives examples of the language. Angluin proposed a new learning model known as *query learning model* (or active learning), where the learner is allowed to *interact* with the teacher, by making questions about the strings of the language. There are different kinds of queries, but the standard combination to be used are: a) *membership queries*: the learner asks if a concrete string belongs to the target language and the teacher answers "yes" or "no"; b) *equivalence queries*: the learner asks if its hypothesis is correct and the teacher answers “yes” if it is correct or otherwise gives a counterexample. According to Angluin’s model, the learner has successfully learnt the target language if it returns the correct hypothesis after asking a finite number of queries.

The learnability of DFA (Deterministic Finite Automata) has been successfully studied in the context of query learning. One of the most important results in this framework was given by D. Angluin (Angluin, 1987). She proved that DFA can be identified in polynomial time using membership and equivalence queries. Later, there were developed more efficient versions of the same algorithm trying to increase the parallelism level, to reduce the number of EQs, etc. (see (Rivest and Schapire, 1993), (Hellerstein et al., 1995), (Balcazar et al., 1997)). Moreover, some new type of queries have been proposed to learn DFA, such as corrections queries, that has led to some interesting results (Becerra-Bonache et al., 2006). Angluin and Kharitonov (Angluin and Kharitonov, 1991) showed that the problem of identifying the class of context-free languages from membership and equivalence queries is computationally as hard as the cryptographic problems. In order to obtain some positive learnability results for classes of languages more powerful than regular, researchers have used different techniques: to investigate subclasses of context-free languages, to give structural information to the learner, to reduce the problem to the learning of regular languages, etc. For more details, see (de la Higuera, 2010).

In Gold’s and Angluin’s model, exact learning is required. However, this has always been considered a hard task to achieve. Based on theses ideas, Valiant introduced the *PAC model*: a distribution-independent model of learning from random examples (Valiant, 1984). According to this model, there exist an unknown distribution over the examples, and the learner receives examples sampled under this distribution. The learner is required to learn under any distribution, but exact learning is not required (since one may be unlucky during the sampling process). A successful learning algorithm is one that with high probability finds a grammar whose error is small.

In the PAC learning model, the requirement that the learning algorithm must learn under any distribution is too hard and has led to very few positive results. Even for the case of DFA, most results are negative. For a review of some positive results in this model, see (de la Higuera, 2010).

# 4 AGENTS AND FORMAL LANGUAGES

Multi-agent systems offer strong models for representing complex and dynamic real-world environments. The formal apparatus of agent technology provides a powerful and useful set of structures and processes for designing and building complex applications. Multi-agent systems promote the interaction and cooperation of autonomous agents to deal with complex tasks. Taking into account that computing languages is a complex task, formal language theory has taken advantage of the idea of formalizing architectures where a hard task is distributed among several task-specific agents that collaborate in the solution of the problem: in this case, the generation/recognition of language.

The first generation of formal grammars, based in rewriting, formalized classical computing models. The idea of several devices collaborating for achieving a common goal has given rise to a new generation of formal languages that form an agent-based subfield of the theory. *Colonies*, *grammar systems* and *eco-grammar systems* are examples of this new generation of formal languages. All these new types of formalisms have been proposed as grammatical models of agent systems.

The main advantage of those agent-based models is that they increase the power of their component grammars thanks to interaction, distribution and cooperation.

## 4.1 Colonies

Colonies as well-formalized language generating devices have been proposed in (Kelemen and Kelemenová, 1992), and developed during the nineties in several directions in many papers (Baník, 1996), (Kelemenová and Csuhaj-Varjú, 1994), (Păun, 1995), (Sosík and Štýbnar, 1997), (Martín-Vide and Păun, 1998), (Martín-Vide and Păun, 1999), ( Kelemenová, 1999), (Sosík, 1999), (Dassow et al., 1993), (Kelemen, 1998). Colonies can be thought of as grammatical models of multi-agent systems motivated by Brooks’ subsumption architectures (Brooks, 1990). They describe language classes in terms of behavior of collections of very simple, purely reactive, situated agents with emergent behavior.

A colony consists of a finite number of simple agents which generate finite languages and operate on a shared string of symbols –the *environment*– without any explicitly predefined strategy of cooperation. Each component has its own reactive behavior which consists in: 1) sensing some aspects of the *context* and 2) performing elementary tasks in it in order to achieve some local changes. The environment is quite passive, its state changes only as result of acts agents perform on its string. Because of the lack of any predefined strategy of cooperation, each component participates in the rewriting of current strings whenever it can participate in it. The behavior of a colony is defined as the set of all the strings which can be generated by the colony from a given starting string.

Colonies offer a formal framework for the emergence of complex behaviors by using purely reactive simple components. The main advantage of colonies is their generative power, the class of languages describable by colonies that make use of strictly regular components is beyond the set describable in terms of individual regular grammars.

In the last decade, computational models have become mostly bio-inspired. In the same way, the basic concept of colony, that is taken first from nature, has been developed by means of several bio-inspired computing theories, giving rise to membrane systems (Păun, 2000), tissue P systems (Martín-Vide et al., 2002) or NEPs (Castellanos et al., 2003). Despite the differences, the main idea of colonies remains in these models: *interaction, collaboration, emergence*. The most relevant contribution of bio-inspired models to the basic formalization seems to be the concept of evolution in the configuration and definition of the components of the system during the computation.

## 4.2 Grammar Systems

Grammar system theory is a consolidated and active branch in the field of formal languages 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 "to demonstrate a particular possibility of studying complex systems in a purely syntactic level" (Csuhaj-Varjú et al., 1994) or, what is the same, to propose a grammatical framework for multi-agent systems.

A grammar system is a set of grammars working together, according to a specified protocol, to generate a language. Note that while in classical formal language theory *one* grammar (or automaton) works individually to generate (or recognize) *one* language, here we have *several* grammars working together in order to produce *one* language.

The theory was launched in 1988 (Csuhaj-Varjú and Dassow, 1990), when Cooperating Distributed Grammar Systems (CDGS) were proposed as a syntactic model of the blackboard architecture of problem solving. A CDGS consists of a finite set of generative grammars with a common sentential form (axiom) that cooperate in the derivation of a common language. Component grammars generate the string in turns (thus, sequentially), under some cooperation protocol. At each moment in time, one grammar (and just one) is active, this is, rewrites the common string, while the rest of grammars of the CDGS are inactive. Conditions under which a component can start/stop its activity on common sentential form are specified in the cooperation protocol. Terminal strings generated in this way form the language of the system.

An analogy can be drawn between CDGS and the blackboard model of problem solving described in (Nii, 1989) as consisting of three major components: 1) *Knowledge sources*. The knowledge needed to solve the problem is partitioned into knowledge sources, which are kept separate and independent; 2) *Blackboard data structure*. Problem solving state data are kept in a global database, the *blackboard*. Knowledge sources produce changes in the blackboard that lead incrementally to a solution to the problem. Communication and interaction among knowledge sources take place solely through the blackboard; 3) *Control*. Knowledge sources respond opportunistically to changes in the blackboard. There is a set of control modules that monitor changes in the blackboard and decide what actions to take next. Criteria are provided to determine when to terminate the process. In CDGS, component grammars correspond to knowledge sources. The common sentential form in CDGS plays the same role as the blackboard data structure. And finally, the protocol of cooperation in CDGS encodes control on the work of knowledge sources. The rewriting of a non-terminal symbol can be interpreted as a developmental step on the information contained in the current state of the blackboard. And, finally, a solution to the problem corresponds to a terminal word.

One year later, in 1989, Parallel Communicating Grammar Systems (PCGS) were introduced as a grammatical model of parallelism (Păun and Sântean, 1989). A PCGS consists of several grammars with their respective sentential forms. In each time unit, each component uses a rule, which rewrites the associated sentential form. Cooperation among agents takes place thanks to the so-called *query symbols* that allow communication among components.

If CDGS were considered a grammatical model of the blackboard system in problem solving, PCGS can be thought of as a formal representation of the *classroom model*. Let us take the blackboard model and make the following modifications: 1) Allow each knowledge source to have its own 'notebook' containing the description of a particular subproblem of a given problem; 2) Allow each knowledge source to operate only on its own 'notebook' and let there exists one distinguished agent which operates on the 'blackboard' and has the description of the problem; 3) and finally, allow agents to communicate by request the content of their own 'notebook'. These modifications on the blackboard model lead to the 'classroom model' of problem solving where the classroom leader (the master) works on the blackboard while pupils have particular problems to solve in their notebooks. Master and pupils can communicate and the global problem is solved through such cooperation on the blackboard. An easy analogy can be established between PCGS and the classroom model: pupils correspond to grammars which make up the system, and their notebooks correspond to the sentential forms. The set or rules of grammars encode knowledge of pupils. The distinguished agent corresponds to the 'master'. Rewriting a nonterminal symbol is interpreted as a developmental step of the information contained in the notebooks. A partial solution, obtained by a pupil corresponds to a terminal word generated in one grammar, while solution of the problem is associated to a word in the language generated by the 'master.'

The sequential CDGS and the parallel PCGS are the two main types of grammar systems. However, since 1988, the theory has developed into several directions, motivated by several scientific areas. Besides distributed and decentralized artificial intelligence, artificial life, molecular computing, robotics, natural language processing, ecology, sociology, etc. have suggested some modifications of the basic models, and have given rise to the appearance of different variants and subfields of the theory. For more information on those new types see (Csuhaj-Varjú et al., 1994) and (Dassow et al., 1997).

### 4.3 Eco-grammar Systems

Eco-grammar systems have been introduced in (Csuhaj-Varjú et al., 1996) and provide a syntactical framework for eco-systems, this is, for communities of evolving agents and their interrelated environment. 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'. Within an eco-grammar system we can distinguish two types of components *environment* and *agents*. Both are represented at any moment by a string of symbols that identifies current state of the component. These strings change according to sets of evolution rules. Interaction among agents and environment is carried out through agents' actions performed on the environmental state by the application of some productions from the set of action rules of agents.

An eco-grammar system can be thought of as a generalization of CDGS and PCGS. If we superpose a CDGS and a PCGS, we obtain a system consisting of grammars that contain individual strings (like in PCGS) and a common string (like in CDGS). If we call this common string *environment* and we mix the functioning of CDGS and PCGS, letting each component to work on its own string and on the environmental string, something similar to an ecosystem is obtained. If we add one more grammar, expressing evolution rules of the environment, and we make evolution of agents depend on the environmental state, the thing we obtain is an eco-grammar system.

The concept of eco-grammar system is based on six postulates formulated according to properties of artificial life (Langton, 1989):

1. An ecosystem consists of an *environment* and a *set of agents*.
2. In an ecosystem there is a *universal clock* which marks time units, the same for all the agents and for the environment, according to which agents and environment evolution is considered.
3. Both *environment* and agents have characteristic *evolution rules* which are in fact L systems (Lindenmayer, 1968; Kari et al., 1997), hence are applied in a parallel manner to all the symbols describing agents and environment; such a (rewriting) step is done in each time unit.
4. *Evolution rules of environment* are *independent* on agents and on the state of the environment itself. Evolution rules of agents *depend on* the state of the environment.
5. Agents act on the environment according to *action rules*, which are pure rewriting rules used sequentially. In each time unit, each agent uses one action rule which is chosen from a set depending on current state of the agent.
6. *Action has priority over evolution* of the environment. At a given time unit exactly the symbols which are not affected by action are rewritten by evolution rules.

## 5 CONCLUSIONS

According to (Weiss, 1993), the interest in multiagent systems is founded on the insight that many real-world problems are best modeled using a set of agents instead of a single agent. Multi-agent modeling makes possible to cope with natural constraints like the limitation of the processing power of a single agent and to profit from inherent properties of distributed systems like robustness, fault tolerance, parallelism and scalability. These properties have facilitated the application of multi-agent technology to many types of systems that help humans to perform several tasks.

Machine learning is one of the core fields of Artificial Intelligence, since Artificial Intelligence has been defined as "the science and engineering of making intelligent machines" and the ability to learn is one of the most fundamental attributes of intelligent behavior. It is usually agreed that a system capable of learning deserves to be called intelligent; and conversely, a system being considered as intelligent is, among other things, usually expected to be able to learn.

Formalization has a long tradition in science, besides traditional fields such as physics or chemistry, other scientific areas such as medicine, cognitive and social sciences and linguistics have shown a tendency towards formalization. The use of formal methods has led to numerous results that would have been difficult to be obtained without such formalization. Formal language theory provides good tools to formalize different problems. This flexibility and abstraction has been proven by the application of formal languages to the fields of linguistics, economic modeling, developmental biology, cryptography, sociology, etc.

From what we have said, it follows that multi-agent systems, machine learning and formal language theory provide flexible and useful tools that can be used in different research areas due to their versatility. All three areas have revealed to be very useful for dealing with complex systems. MAS provide principles for the construction of complex systems and mechanisms for coordination. Formal language theory offers mathematical tools to formalize complex systems. And machine learning techniques help to deal with the complexity of complex systems by endowing agents with the ability of improving their behavior. We have seen in this paper that some intersection between those areas has been performed: agents with learning, agents with formal languages and formal languages with learning.

Future research should help to further integrate the three fields considered in this paper in order to obtain what in (Huhns and Weiss, 1998) is seen as a must: a *formal theory of multi-agent learning.*

Another important and challenge working direction is the application of this formal theory of multi-agent learning to a real world domain as is the area of processing natural language. The interaction between researchers in those three topics can provide good techniques and methods for improving our knowledge about how languages are processed. The advances in the area of natural language processing may have important consequences in the area of artificial intelligence since they can help the design of technologies in which computer will be integrated into the everyday environment, rendering accessible a multitude of services and applications through easy-to-use human interfaces.

## ACKNOWLEDGEMENTS

The work of Leonor Becerra-Bonache has been supported by Pascal 2 Network of Excellence. The work of M. Dolores Jiménez-López has been supported by the Spanish Ministry of Science and Innovation under the Coordinated Research Project TIN2011-28260-C03-00 and the Research Project TIN2011-28260-C03-02.



## REFERENCES

Angluin, D. (1987). Learning Regular Sets from Queries and Counterexamples. *Information and Computation*, 75, 87–106.

Angluin, D. and Kharitonov, M. (1991). When Won't Membership Queries Help? In *STOC'91: 24th Annual ACM Symposium on Theory of Computing* (pp. 444–453). New York: ACM Press.

Balcázar, J.L., Díaz, J., Gavaldà, R. and Watanabe, O. (1997). Algorithms for Learning Finite Automata from Queries: A Unified View. In Du, D.Z. and Ko. K.I. (Eds.), *Advances in Algorithms, Languages, and Complexity* (pp. 73–91). Dordrech: Kluwer Academic Publishers.

Baník, I. (1996). Colonies with Position. *Computers and Artificial Intelligence*, 15, 141–154.

Becerra-Bonache, L., Dediu, A.H. and Tirnauca, C. (2006). Learning DFA from Correction and Equivalence Queries. In Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T. and Tomita, E. (Eds.), *ICGI 2006* (pp. 281–292). Heidelberg: Springer.

Brooks, R.A. (1990). Elephants don't Play Chess. *Robotics and Autonomous Systems*, 6, 3–15.

Castellanos, J., Martín-Vide, C., Mitrana, V. and Sempere, J.M. (2003). Networks of Evolutionary Processors. *Acta Informatica*, 39, 517–529.

Clark, A. (2004). Grammatical Inference and First Language Acquisition. In *Workshop on Psychocomputational Models of Human Language Acquisition* (pp. 25–32). Geneva.

Csuhaj-Varjú, E. and Dassow, J. (1990). On Cooperating/Distributed Grammar Systems. *Journal of Information Processing and Cybernetics (EIK)*, 26, 49–63.

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

Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A. and Păun, Gh. (1996). Eco-Grammar Systems: A Grammatical Framework for Life-Like Interactions. *Artificial Life*, 3(1), 1–28.

Dassow, J., Kelemen, J. and Păun, Gh. (1993). On Parallelism in Colonies. *Cybernetics and Systems*, 14, 37–49.

Dassow, J., Păun, Gh. and Rozenberg, G. (1997). Grammar Systems. In Rozenberg, G., Salomaa, A. (Eds.), *Handbook of Formal Languages* (Vol. 2, pp. 155-213). Berlin: Springer.

de la Higuera, C. (2010). *Grammatical Inference: Learning Automata and Grammars*. Cambridge: Cambridge University Press.

Fernau, H. and de la Higuera, C. (2004). Grammar Induction: An Invitation to Formal Language Theorists. *Grammars*, 7, 45–55.

Gold, E.M. (1967). Language identification in the limit. *Information and Control*, 10, 447–474.

Hellerstein, L., Pillaipakkamnatt, K., Raghavan, V. and Wilkins, D. (1995). How many Queries are Needed to Learn?. In *27th Annual ACM Symposium on the Theory of Computing* (pp. 190–199). ACM Press.

Huhns, M. and Weiss, G. (1998). Guest Editorial. *Machine Learning*, 33, 123–128.

Jedrzejowicz, P. (2011). Machine Learning and Agents. In O'Shea, J. et al. (Eds.), *KES-AMSTA 2011* (pp. 2–15). Berlin: Springer.

Kari, L., Rozenberg, G. and Salomaa, A. (1997). L Systems. In Rozenberg, G. and Salomaa, A. (Eds.), *Handbook of Formal Languages* (Vol. 1, pp. 253–328). Berlin: Springer.

Kelemen, J. (1998). Colonies -A Theory of Reactive Agents. In Kelemenová, A. (Ed.), *Proceedings on the MFCS'98 Satellite Workshop on Grammar Systems* (pp. 7–38). Opava: Silesian University.

Kelemen, J. and Kelemenová, A. (1992). A Grammar-Theoretic Treatment of Multiagent Systems. *Cybernetics and Systems*, 23, 621–633.

Kelemenová, A. (1999). Timing in Colonies. In Păun, Gh. and Salomaa, A. (Eds.), *Grammatical Models of Multi-Agent Systems*. London: Gordon and Breach.

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

Langton, Ch. (1989). Artificial Life. In Langton, Ch. (Ed.), *Artificial Life* (pp. 1–47). California: Addison-Wesley.

Lindenmayer, A. (1968). Mathematical Models for Cellular Interaction in Development, I and II. *Journal of Theoretical Biology*, 18, 280–315.

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

Martín-Vide, C. and Păun, Gh. (1999). New Topics in Colonies Theory. *Grammars*, 1, 209–223.

Martín-Vide, C., Păun, Gh., Pazos, J. and Rodríguez-Patón, A. (2002). Tissue P Systems. *Theoretical Computer Science*, 296(2), 295–326.

Nii, H.P. (1989). Blackboard Systems. In Barr, A., Cohen, P.R. and Feigenbaum, E.A. (Eds.), *The Handbook of Artificial Intelligence* (Vol. IV, pp. 3–82). California: Addison-Wesley.

Nilsson, N. J. (1998). *Introduction to Machine Learning. An Early Draft of a Proposed Textbook*, http://robotics.stanford.edu/people/nilsson/mlbook.html.

Panait, L. and Luke, S. (2005). Cooperative Multi-Agent Learning: The State-of-the-Art. *Autonomous Agents and Multi-Agent Systems*, 11(3), 387–434.

Păun, Gh. (1995). On the Generative Power of Colonies. *Kybernetika*, 31, 83–97.

Păun, Gh. (2000). Computing with Membranes. *Journal of Computer and Systems Sciences*, 61(1), 108–143.

Păun, Gh. and Sântean, L. (1989). Parallel Communicating Grammar Systems: The Regular Case. *Annals of the University of Bucharest*, 38, 55–63.

Rivest, R.L. and Schapire, R.E. (1993). Inference of Finite Automata Using Homing Sequences. *Information and Computation*, 103(2), 299–347.

Rozenberg, G. and Salomaa, A. (Eds.) (1997). *Handbook of Formal Languages*. Berlin: Springer.

Sen, S. and Weiss, G. (1999). Learning in Multiagent Systems. In Weiss, G. (Ed.), *Multiagent Systems* (pp. 259-298). Cambridge: MIT Press.

Shoham, Y., Powers, R. and Grenage, T. (2007). If Multi-Agent Learning is the Answer, What is the Question? *Artificial Intelligence*, 171(7), 365–377.

Sosík, P. (1999). Parallel Accepting Colonies and Neural Networks. In Păun, Gh. and Salomaa, A. (Eds.), *Grammatical Models of Multi-Agent Systems*. London: Gordon and Breach.

Sosík, P. and Štýbnar, L. (1997). Grammatical Inference of Colonies. In Păun, Gh. and Salomaa, A. (Eds.), *New Trends in Formal Languages*. Berlin: Springer.

Stone, P. and Veloso, M. (2000). Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8, 345–383.

Valiant, L.G. (1984). A Theory of the Learnable. *Communication of the ACM*, 27, 1134–1142.

Weiss, G. (1993). Learning to Coordinate Actions in Multi-Agent Systems. In *Proceedings of the 13th International Conference on Artificial Intelligence* (pp. 311–316).

Weiss, G. (1998). A Multiagent Perspective of Parallel and Distributed Machine Learning. In *Proceedings of the 2nd International Confernce on Autonomous Agents* (pp. 226–230).

Weiss, G. and Dillenbourg, P. (1998). What is "Multi" in Multi-Agent Learning?. In Dillenbourg, P. (Ed.), *Collaborative Learning: Cognitive and Compuational Approaches* (Chapter 4). Oxford: Pergamon Press.