L Systems as Bio-MAS for Natural Language Processing

Leonor Becerra-Bonache, Suna Bensch, and M. Dolores Jiménez-López

Abstract. In this paper, we claim that Lindenmayer systems (L systems) –more precisely, ET0L systems– can be considered as *bio-inspired multi-agent systems* that, because of its inherent features, can be usefully applied to the field of natural language processing (NLP). L systems are a biologically inspired branch of the field of formal languages that provide a *parallel* and non-sequential grammatical formalism and that can be expressed as a multi-agent system. Taking into account these features and the benefits of the multi-agent approach to NLP, we propose to apply L systems to the description, analysis and processing of natural languages.

1 Introduction

In general, multi-agent systems offer strong models for representing complex and dynamic real-world environment. 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 and useful set of structures and processes for designing and building complex software applications. 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. The metaphor

---

Leonor Becerra-Bonache
Research Group on Mathematical Linguistics, Universitat Rovira i Virgili,
43002 Tarragona, Spain
e-mail: `leonor.becerra@urv.cat`

Suna Bensch
Institutionen för Datenvetenskap, Umeå Universitet, 90187 Umeå, Sweden
e-mail: `suna@cs.umu.se`

M. Dolores Jiménez-López
Research Group on Mathematical Linguistics, Universitat Rovira i Virgili,
43002 Tarragona, Spain
e-mail: `mariadolores.jimenez@urv.cat`

Y. Demazeau et al. (Eds.): Trends in PAAMS, AISC 71, pp. 395–402.
springerlink.com

of autonomous problem solving entities cooperating and coordinating to achieve their objectives is a natural way of conceptualizing many problems. The multi-agent system literature spans a wide range of fields including robotics, mathematics, linguistics, psychology, and sociology, as well as computer science. Multi-agent systems promote the interaction and cooperation of autonomous agents in order to deal with complex tasks. This architecture has the advantage of distributing a hard task among several task-specific agents that collaborate in the solution of the problem. In a multi-agent system, agents cooperate to achieve a common objective. This idea seems to be very adequate in the field of natural language processing. In fact, the task of processing natural language requires multiple agents working together in the pursuit of a common goal: the *generation/understanding of language*. Some authors have suggest the possibility of using multi-agent systems in natural language processing to represent cooperation among distinct linguistic levels [10, 6, 17, 16, 12, 2]. With the idea of defending the adequacy of multi-agent systems in the field of natural language processing, we propose the application of Lindenmayer Systems to the description, analysis and processing of natural language. L systems can be seen as a biologically inspired branch of the field of formal languages that provide a *parallel* and non sequential grammatical formalism. In this paper, we claim that L systems –more precisely, ET0L systems– can be considered as *bio-inspired multi-agent systems* that, because of its inherent features, can be usefully applied to the field of natural language processing.

The paper is organized as follows. Section 2 introduces Lindenmayer systems and the basic definition of ET0L systems. Section 3 expresses L systems as a bio-inspired multi-agent systems. Section 4 presents the application of L systems to natural language processing. Final conclusions are presented in section 5. Throughout the paper, we assume the reader to be familiar with basic notions in the theory of formal languages. For more information the reader is referred to [15].

2 Lindenmayer Systems

Aristid Lindenmayer introduced, in 1968 [13], specific rewriting systems as models of developmental biology, which today are called Lindenmayer systems or L systems. L systems model biological growth and because growth happens in multiple areas of an organism, growth is parallel. The essential difference between Chomsky grammars and L systems lies in the method of applying productions. In Chomsky grammars productions are applied sequentially, whereas in L systems they are applied in parallel and simultaneously replace all letters in a given word. This difference reflects the biological motivation of L systems. Productions are intended to capture cell divisions in multicellular organisms, where many divisions may occur at the same time. The investigations of L systems are an important and wide area in the theory of formal languages. The modelling of different environmental influences, for example, growth during day versus night, lead to different L systems and thus to different L languages. The study of L languages has resulted in a language hierarchy, namely the L system hierarchy. Lindenmayer systems are well investigated parallel rewriting systems. For an overview see [11] and for the mathematical theory of L systems see [14].

Definition 1. *An extended tabled Lindenmayer system without interaction (ET0L system, for short) is a quadruple $G = (\Sigma, H, \omega, \Delta)$, where Σ is the alphabet, Δ is the terminal alphabet, $\Delta \subseteq \Sigma$, H is a finite set of finite substitutions from Σ into Σ^*, and $\omega \in \Sigma^*$ is the axiom.*

Definition 2. *For x and y in Σ^* and $h \in H$, we write $x \underset{h}{\Longrightarrow} y$ if and only if $y \in h(x)$. A substitution h in H is called a* table.

Definition 3. *The language generated by G is defined as:*

$L(G) = \{w \in \Delta^* \mid \omega \underset{h_{i_1}}{\Longrightarrow} w_1 \underset{h_{i_2}}{\Longrightarrow} \ldots \underset{h_{i_m}}{\Longrightarrow} w_m = w$ *for some* $m \geq 0$ *and* $h_{i_j} \in H$ *with* $1 \leq j \leq m\}$.

By $\mathscr{L}$(ET0L) we denote the family of ET0L languages.

Example 1. Let $G_1 = (\{A,B,C,a,b,c\}, \{h_1,h_2\}, ABC, \{a,b,c\})$ be an ET0L system, where h_1 and h_2 are given as follows:

$$h_1 = \{A \to aA, B \to bB, C \to cC, a \to a, b \to b, c \to c\},$$
$$h_2 = \{A \to a, B \to b, C \to c, a \to a, b \to b, c \to c\}.$$

The axiom ABC can be rewritten using the first table h_1 or the second table h_2. Using the first three productions $A \to aA$, $B \to bB$, $C \to cC$ in h_1 adds a symbol a, b, and c, respectively in every derivation step. Using table h_2 terminates the derivation process. Consider the derivation of the word $a^2b^2c^2$:

$$ABC \underset{h_1}{\Longrightarrow} aAbBcC \underset{h_2}{\Longrightarrow} aabbcc$$

The language generated is $L(G_1) = \{a^nb^nc^n \mid n, m \geq 1\}$.

Example 2. Let $G_2 = (\{A,B,a,b\}, \{h_1,h_2,h_3,h_4\}, AA, \{a,b\})$ be an ET0L system, where the tables are given as follows:

$$h_1 = \{A \to aA, B \to B, a \to a, b \to b\},$$
$$h_2 = \{B \to A, A \to B, a \to a, b \to b\},$$
$$h_3 = \{B \to bB, A \to A, a \to a, b \to b\},$$
$$h_4 = \{A \to a, B \to b, a \to a, b \to b\}.$$

Table h_1 introduces a symbol a in every derivation step and table h_3 introduces a symbol b in every derivation step. Table h_2 serves as switch between the symbols A and B and table h_4 terminates the derivation process. Consider the derivation for the word *baba*:

$$AA \underset{h_2}{\Longrightarrow} BB \underset{h_3}{\Longrightarrow} bBbB \underset{h_2}{\Longrightarrow} bAbA \underset{h_4}{\Longrightarrow} baba$$

The language generated is $L(G_2) = \{ww \mid w \in \{a,b\}^+\}$.

Example 3. Let $G_3 = (\{A,B,C,D,a,b,c,d\},\{h_1,h_2,h_3,h_4\},ABCD,\{a,b,c,d\})$ be an ET0L system, where the tables are given as follows:

$$h_1 = \{A \to aA, C \to cC, B \to B, D \to D, a \to a, b \to b, c \to c, d \to d\},$$
$$h_2 = \{A \to a, C \to c, B \to B, D \to D, a \to a, b \to b, c \to c, d \to d\},$$
$$h_3 = \{B \to bB, D \to dD, A \to A, C \to C, a \to a, b \to b, c \to c, d \to d\},$$
$$h_4 = \{B \to b, D \to d, A \to A, C \to C, a \to a, b \to b, c \to c, d \to d\}.$$

Using tables h_1 or h_3 introduce in every derivation step the symbols a and c or the symbols b and d, respectively. The tables h_2 and h_4 are used in order to terminate the derivation process. The language generated is $L(G_3) = \{a^n b^m c^n d^m \mid n,m \geq 1\}$.

3 L Systems as Bio-inspired Multi-agent Systems

Lindenmayer systems may be interpreted in terms of multi-agent systems. ET0L systems define systems of distributed components in which components can be viewed as autonomous problem solvers that must collaborate in order to perform complex tasks. Every table h in an L system can be seen as an agent that has a particular concern. Each agent has its rules and participate in the solution of a complex problem: the generation of the language.

Ferber [5] defines a multiagent system as having the following basic entities. Those basic entities can be interpreted in terms of L Systems as follows:

- An *environment*. In our system, the environment where the agents collaborate is the *string* they are rewriting (starting by the axiom).
- A *set of objects* that exist in the environment. The objects in the environment are *symbols* over the alphabet.
- A *set of agents*. The agents in ET0L systems are *the tables h in H*. Each table h is an agent that has a particular concern.
- A *set of operations* that agents can use to sense and affect objects. The set of operations in L systems are the *rules* each table h has that allow them to participate in the solution of the problem.
- A set of *universal laws* that define the reaction of the environment to agent operations. In our case, the reaction of the environment to agent operations is just the *modification of the string* due to the rules applied by the agents.

According to Wooldridge [19], agents in a multi-agent systems have three important characteristics:

- *Autonomy*: the agents are at least partially autonomous.
- *Local views*: no agent has a full global view of the system.
- *Decentralization*: there is no designated controlling agent.

In L Systems, agents h are *autonomous* since each of them has its set of rules that are independent on the rules other agents in the system have. Agents have local views, since no agent alone can solve the problem of generating the language of the system, but collaboration between all the agents in the system is needed in order to generate the language. And, of course, no agent in L systems controls the whole process.

From what we have said, it follows that L systems can be considered as multi-agent systems. Moreover, Lindenmayer systems can be seen as *bio-inspired* multi-agent systems. In fact, L systems are the first bio-inspired model in the field of formal language theory. Aristid Lindenmayer introduced L systems in 1968 as a theoretical framework in order to model the development of filamentous organisms, which are composed of cells. These cells receive inputs from their neighbours and change their states and produce outputs based on their states and the input received. Cell division is modelled by inserting two new cells in the filament in order to replace one cell. With these theoretical framework of Lindenmayer an organism as a whole was provided that models individual acts of division, unequal divisions, interaction of two or more cells and cell enlargement. The possible combinations of interactions among more than a handful of cells becomes rapidly unmanagable without a mathematical theory and computer application, which are provided both in the framework of Lindenmayer systems.

4 Applying L Systems to Natural Language Processing

> 'There are different ways to organize a generation system to gain more flexibility. The ideal organization lets each module contribute the choices to do with its area concern, without having to make other decisions that would be better made by some other module.' [1, p. 504].

The above quotation clearly defends as ideal organization in *natural language processing* the multi-agent one. To consider a distributed structure, where several agent components have limited information and operate in an independent way, may facilitate very much the task of processing natural language. In general, formal and computational approaches to natural language demand non-hierarchical, parallel, distributed models in order to explain the complexity of linguistic structures as the result of the interaction of a number of independent but cooperative components. In natural language processing, researchers have turned from the initial serial models to highly parallel ones. *Distribution* and *cooperation* are two important notions in natural language processing. In this field, it is usual to deal with complex tasks *distributed* among a set of 'processors' that *work together* in a well defined way. In a distributed/cooperative architecture for natural language processing, agents can work together to solve problems that are beyond their individual capabilities. Each agent in the system can work independently, but problems faced by each of them cannot be completed without cooperation. Cooperation is necessary because no agent has sufficient expertise, resources and information to solve the problem. The suitability shown by cooperative distributed architectures, have led to apply distributed models to the field of natural language processing. In text analysis, for example, we can find models as *TALISMAN* [17] defined as a distributed architecture for text analysis in French, that includes linguistic agents that correspond either to classical levels in linguistics (morphology, syntax, semantics) or to complex language phenomena. The main goal of this model is to show that complex linguistic phenomena can be defined and processed in a distributed architecture. The

CARAMEL model proposed in [7] is also an example of distributed architecture in natural language understanding. The model proposed by those authors intends to be applied to several different topics such as text understanding, dialogue management, making abstracts or intelligent tutoring systems.

The idea of distribution present in multi-agent systems is adequate for every component of a grammar. In [3], a highly distributed organization of *syntax* where components are determined by representations they recover, is suggested. The author defends four modules in the syntactic processor, each related with a 'representational' or 'informational' aspect of grammar: Phrase Structure; Chains; Thematic Structure; Coindexation (Binding and Control Theory). Another example of distribution, this time in *semantic and phonological* modules, is presented in [9] where the author suggests that: '*... meaning, like phonological structure, is organized into independent but interacting tiers, each of which contributes a different class of conceptual distinctions to meaning as a whole.*' [9, p. 2]. Studies on modularity in *morphology* can be found in [4], for example. About the internal modularity of *pragmatics* some approaches are presented in [18] and [8]. So, from all those examples it follows that not only a grammar as a whole can be regarded as a modular system composed by several parallel interacting components (syntax, semantics, phonology, etc.), but that also every component that builds up grammar can be viewed as *internally distributed*, being divided into several different agents as well.

From what we have said above, it follows that multi-agent systems offer the necessary flexibility in natural language processing. The aim of a multi-agent system for natural language processing is to define a set of agents with different skills that participate in processing of natural language. L systems can be seen as multi-agent systems that offer natural language processing a good tool to deal with the complex task of generate/understanding natural language.

Moreover, L systems present another interesting feature to be applied to natural language processing: *bio-inspiration*. The biological inspiration of L systems is an interesting property from a linguistic point of view since in the last decades there seems to be a tendency to use bio-inspired models in the description and processing of natural languages. Linguistics has not been able to solve the problem of generation/understanding natural language, partly because of the fail in the models adopted. Bio-inspired models could be a possible solution since one of the advantages of such kind of models is to offer more 'natural' tools than the ones used so far. Rewriting methods used in a large number of natural language approaches seem to be not very adequate, from a cognitive perspective, to account for the processing of language.

In the following we sketch one idea of how ET0L systems could be applied to deal with different linguistical levels. We consider the syntactical and the semantical level and thus an ET0L system with two tables h_{syn} and h_{sem}, where h_{syn} deals with syntax and h_{sem} with semantics. The table h_{syn} checks whether a given sentence is syntactically correct and the table h_{sem} assigns the semantical structure. The example is simplified in order to illustrate better the applicability of L systems for natural language processing. The number of the linguistical levels and the theories for these linguistical levels can be choosen as liked. Let h_{syn} be the set of rules

$\{S \to N_p\ V_p, N_p \to Susan, V_p \to V\ N_p, Agent \to Agent, V \to kissed, N_p \to John\}$ and h_{sem} be the set of rules $\{Susan \to Agent, V \to V, N_p \to N_p, kissed \to Action, John \to Patient, Agent \to Agent\}$. Let S be the axiom. Table h_{syn} rewrites the axiom to $N_p\ V_p$. Now again only table h_{syn} is applicable and rewrites all symbols (namely $N_p\ V_p$) to $Susan\ V\ N_p$. At this stage table h_{syn} is no more applicable but table h_{sem} is and rewrites all symbols into $Agent\ V\ N_p$. Applying table h_{syn} again yields $Agent\ kissed\ John$. Finally table h_{sem} is applied and rewrites all symbols into $Agent\ Action\ Patient$. In this simplified example one table assigns a syntactical structure to a given sentence and as soon as a semantical category can be assigned to a syntactical category this is done by the other table.

5 Conclusions

Formal and computational approaches of natural language demand non-hierarchical, parallel, distributed models in order to explain the complexity of linguistic structures as the result of the interaction of a number of independent but cooperative modules. Natural language processing needs frameworks that are able to generate/understand the structures present in natural languages with simple mechanisms that describe/explain those structures in a more natural way than the usual rewriting systems. In natural language processing, researchers have turned from the initial serial models to highly parallel ones. In theoretical linguistics, the hierarchical view of grammar has revealed as problematic and there has been a search for systems with parallel and autonomous components that cooperate in order to generate natural language. Since languages, either natural or artificial, are particular cases of symbol systems and the manipulation of symbols is the stem of formal language theory, it seems adequate to look for bio-inspired models that have been defined in that research area. Lindenmayer systems are the first bio-inspired model proposed in the field of formal languages and it is also the first one that replaces the sequential rewriting by the parallel one. Therefore, it seems that Lindenmayer systems offer a great deal of the properties that seem to be necessary in order to approach linguistic structures. Moreover, it has been pointed out that using a multi-agent system for natural language processing has several advantages and, as we have showed in this paper, L systems can be considered as multi-agent systems. Models in natural language processing demand *bio-inspired devices* that avoid the *sequential* rewriting, that have *enough expressiveness* to describe natural languages and that present a *distributed* architecture. L systems are described as a *parallel* and non sequential grammatical formalism, they are *biologically* inspired, they can generate the *non-context free structures* present in natural language and can offer a distributed/cooperative architecture. Therefore, we consider that it could be interesting to apply L systems to the description/processing of natural language in order to see if this parallel distributed bio-inspired model may improve current NLP approaches.

Acknowledgements. The research of Leonor Becerra-Bonache was supported by a Marie Curie International Fellowship within the 6^{th} European Community Framework Programme.

References

1. Allen, J.F.: Natural Language Understanding. The Benjamin/Cummings Publishing Company, California (1987)
2. Aref, M.M.: A Multi-Agent System for Natural Language Understanding. In: KIMAS 2003, October 1-3 (2003)
3. Crocker, M.W.: Multiple Meta-Interpreters in a Logical Model of Sentence Processing. In: Brown, C.G., Koch, G. (eds.) Natural Language Understanding and Logic Programming, III, pp. 127–145. Elsevier Science Publishers B.V., North-Holland (1991)
4. Everaert, M., Evers, A., Hybreqts, R., Trommelent, M. (eds.): Morphology and Modularity: In Honour of Henk Schultink. Publications in Language Sciences 29, Foris (1988)
5. Ferber, J.: Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison-Wesley, Harlow (1999)
6. Fum, D., Guida, G., Tasso, C.: A Distributed Multi-Agent Architecture for Natural Language Processing. In: Proceedings of COLING 1988, Budapest, vol. II, pp. 22–27 (1988)
7. Grau, B., Sabah, G., Vilnat, A.: Control in Man-Machine Dialogue. Think 3, 32–55 (1994)
8. Horn, L.R.: Pragmatic Theory. In: Newmeyer, F. (ed.) Linguistic Theory: Foundations, Linguistics: The Cambridge Survey, vol. I, pp. 113–145. Cambridge University Press, Cambridge (1988)
9. Jackendoff, R.: Semantic Structures. MIT Press, Cambridge (1990)
10. Jiménez-López, M.D.: A Grammar Systems Approach to Natural Language Grammar. Linguistics and Philosophy 29, 419–454 (2006)
11. Kari, L., Rozenberg, G., Salomaa, A.: L systems. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 1. Springer, Berlin (1997)
12. Lima, V.L.S., Abrahao, P.R.C., Santa Maria Filho, I.: Distributing linguistic knowledge in a multi-agent natural language processing system: re-modelling the dictionary. Procesamiento del lenguaje natural 23, 104–109 (1998)
13. Lindenmayer, A.: Mathematical models for cellular interaction in development. Journal of Theoretical Biology 18, 280–315 (1968)
14. Rozenberg, G., Salomaa, A.: The Mathematical Theory of L-Systems. Academic Press, New York (1980)
15. Rozenberg, G., Salomaa, A.: Handbook of Formal Languages. Springer, Berlin (1997)
16. Silva, J.L., Lima, V.L.S.: An alternative approach to lexical categorial disambiguation using a multi-agent systems architecture. In: Proceedings of the RANLP 1997, Bulgaria, pp. 6–12 (1997)
17. Stefanini, M.H., Berrendoner, A., Lallich, G., Oquendo, F.: TALISMAN: un système multi-agents gouverné par des lois linguistiques pour le traitement de la langue naturelle. In: Proceedings of the COLING 1992, Nantes, pp. 23–28 (1992)
18. Wilson, D., Sperber, D.: Pragmatics and Modularity. In: Davis, S. (ed.) Pragmatics. A Reader, pp. 583–594. Oxford University Press, New York (1991)
19. Wooldridge, M.: An Introduction to Multi Agent Systems. John Wiley & Sons Ltd., New York (2002)