Communication Network of Symbolic Grammar Systems

TAKASHI HASHIMOTO [*] and TAKASHI IKEGAMI [†]

[*] The Graduate School of Science and Technology, Kobe University

[†] Institute of Physics, College of Arts and Sciences, University of Tokyo

Abstract - Interacting agents with symbolic grammar are proposed in order to study the evolution of computational ability of agents. The algorithmic evolution of the formal grammar system is characterized by Chomsky's hierarchy[1]. Agents with a higher grammar can speak/recognize many more words than those with a lower one. However, when agents form a network, the higher Chomsky grammar is not always advantageous. It is shown that to speak/recognize commonly used words is more favorable in a network.

INTRODUCTION

In this paper we present an evolutionary model of interacting agents with symbolic grammar. Our main concern is to see how a higher level of grammar evolves from the lower ones.

Regarding computational ability as a measure of evolution, we naturally ask, what evolutionary dynamics can elaborate computational ability? Introducing an ensemble of agents, we discuss its evolution. We see evolutionary pathways of climbing up Chomsky's hierarchy and show how they are avoided by the ensemble.

DYNAMICAL MODELING OF COMMUNICATION NETWORK

We characterize each agents by a set of rewriting rules, $V \rightarrow a$, where V is a non-terminal symbol and a is a list of non-terminal and terminal symbols. To generate sentences, the agents apply their rules from the left to right hand side. If there are several rewriting paths, one of them is selected randomly. To recognize sentences, rules are applied in the opposite direction. If a agent rewrites a sentence to the start symbol "S" in finite rewriting steps, the agent can recognize it.

When we make a network of agents, each agent speaks in turn by using its grammar and tries to recognize the sentences spoken by the others.

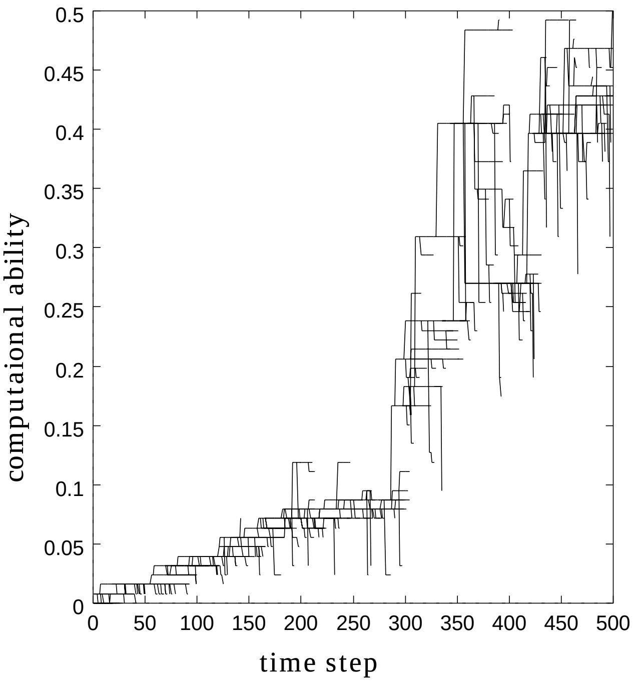
[*] e-mail: toshiwo@sacral.c.u-tokyo.ac.jp

[†] e-mail: ikeg@sacral.c.u-tokyo.ac.jp

Each agent is ranked by the following factors. The agent receive a higher score if it can **a) speak less spoken and longer sentences**, **b) speak less acceptable sentences**, and **c) recognize sentences by less computational time**. The higher scores agents can get, the more likely they are to stay in the network. Agents who get lower scores are replaced by the newly generated agents.

A new agent evolves by the following three mutation processes. a) adding a new rule to a rule list, b) replacing a rule with a new rule, and c) deleting a rule from a rule list.

Each rewriting rule belongs either to a sentence coding rule or a word coding rule. The former is defined as a rule which has non-terminal symbols on its right hand side(r.h.s). On the other hand, the latter contains only terminal symbols on its r.h.s. For example, an agent with grammar, $S{\rightarrow}0X, X{\rightarrow}01$, recognizes ”001” by a sequential change of ”001” into ”S” as $001{\Rightarrow}0X{\Rightarrow}S$. We say that the agent recognizes ”001” as a sentence. If an agent has a direct transforming rule, $S{\rightarrow}001$, he can immediately recognizes the word ”001” as $001{\Rightarrow}S$. We say the agent recognizes ”001” as a word. If a list of symbols is recognized as a word, fewer rewriting steps are required than to recognize it as a sentence.



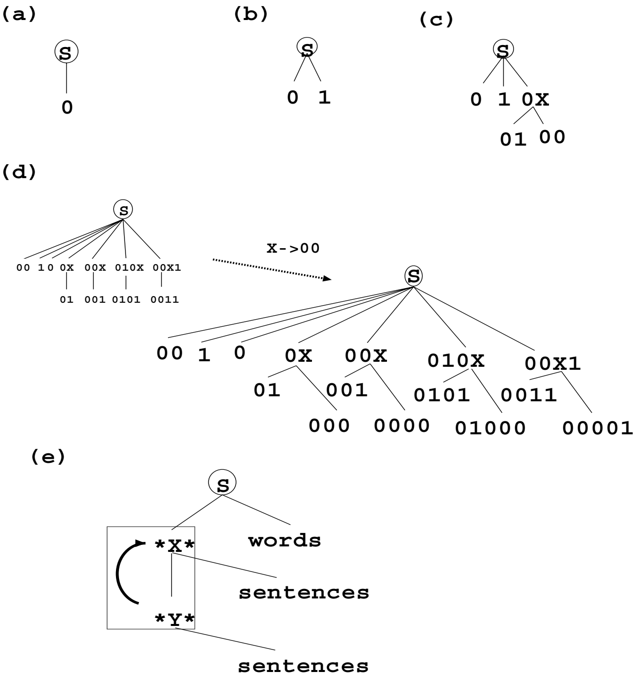


Fig. 1 A phylogeny of grammar is expressed here by depicting agent’s computational ability as a function of time. Each line connects one agent to one of its offspring. It branches off by the mutations. A line is terminated when the corresponding agent is removed.

Fig. 2 An example of algorithmic evolution: These figures display one’s grammar structure as a form of production tree. An initial grammar structure evolves through (a) to (e). (a)sequential structure. (b)branch structure. (c)multi-branch structure. (d)module type evolution. (e)loop structure (shown by a box) See the details in the text.

ALGORITHMIC EVOLUTION

Fig. 1 is an example of evolution from the initial network of agents with randomly generated rules. In terms of Chomsky's hierarchy, they are categorized as a type 3 grammar. Computational ability of an agent is measured by the ratio of speakable sentences to the total number of possible sentences.

The grammar of an agent structurally changes several times and computational ability evolves. We call these changes of grammar algorithmic evolution. At first, a grammar has only a sequential structure (Fig. 2(a)). It is developed into a branch structure (Fig. 2(b)), then a multi-branch structure appears (Fig. 2(c)). The computational ability of the grammar are rather low, since they only speak/recognize finite words.

At the second stage (around time step 192), a new agent with higher ability has emerged by acquiring a module rule, which is defined as a rule that can be utilized by many other rules. A module type evolution through this new rule is drawn in Fig. 2(d).

The third stage (time step 310), a grammatically new agent invades the population. Since the new agent has a loop structure, it can generate a potentially infinite number of sentences. A grammar with a loop structure is called a type 2 grammar. Type 2 grammar can recursively generate sentences (Fig. 2(e)). No type 3 grammar can generate the same set of sentences.

COMMON WORD ENSEMBLE

An agent with the highest grammar level is sometimes removed from the population. It is explained by the emergence of a common word ensemble (CWE), whose members speak/recognize the same set of words. An agent that can't speak/recognize common words becomes disadvantaged. An agent that takes too much time to recognize such words is also removed from the population. What is required is to recognize not a lot of words but to speak/recognize common words within short steps. We can also make a Minimal-All-mighty that is the agent can speak/recognize all sentences with the least number of rules. If a population consists of Minimal-All-Mighties average score becames low. Because a MAM tries to speak shorter sentences. This explains why the population is not occupied by MAMs.

An algorithmic evolution is suppressed by the CWE. However, an algorithmic evolution is not entirely terminated by the appearance of CWE. After all agents become able to speak/recognize common words, agents again have to speak new sentences. Hence evolution occurs again.

In Fig. 3 we can see a punctuated equilibrium. If we plot the ratio of the number of recognizable sentences to the number of rules (top graph), they evolve in a stepwise manner. A set of word coding rules and a set of sentence coding rules evolve in turn. At the initial 300th step the number of word coding rules increases. Then sentence coding rules increase, and a new CWE may evolve. Each rapid increase in the top graph indicates appearance of sentence coding rules. After a new CWE is established the same scenario can be applied.

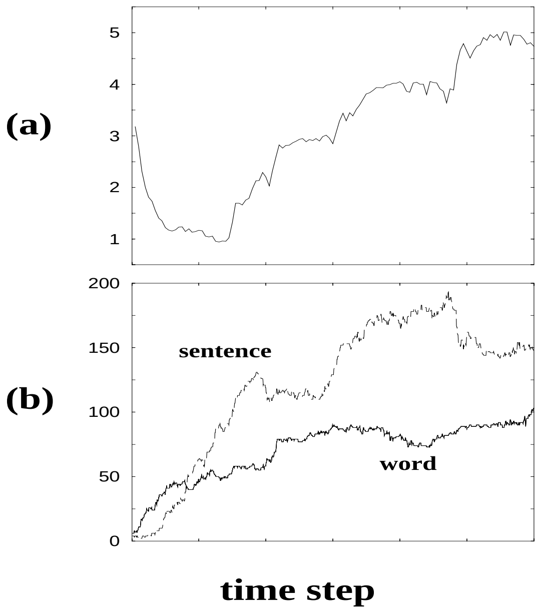


Fig. 3 These graphs are time v.s. (a) the ratio of the number of recognized sentences to the number of rules. (b) the number of sentence coding rules and word coding rules.

CONCLUSION

We have discussed the evolutionary pathway of symbolic grammar in a network. Besides an algorithmic evolution, it is shown that a common word ensemble can emerge. In CWE, each agent tries to speak/recognize common words than to speak rare words. In order to speed up a recognizing process, each agent tries to use word coding rules. When all agents comes to speak/recognize a set of common words, agents starts to speak/recognize long and rarely spoken sentences again. Hence sentence coding rules become useful. This scenario is periodically applied, showing a stepwise evolution.

In conclusion, we stress the importance of social nature of grammar systems. Without such an ensemble structure higher grammars in Chomsky's hierarchy may always evolve.

The edge of chaos is sometimes discussed as a goal of evolution in general. Indeed, there are several simulations showing the advantages of the edge of chaos[2−4]. Since a grammar system is defined on discrete symbols, it is difficult to relate our system to the idea of the edge of chaos. Also a grammar system has essential undecidability, no dynamical system can construct grammar system in a strict sense. However one approach is shown by Crutchfield[3] through symbolic dynamics and ϵ-machines. More details will be discussed in Ref. 5.

REFERENCES

1. J. E. Hopcroft and J. D. Ullman *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, Massachusetts (1979).
2. K. Kaneko and J. Suzuki. Evolution to the Edge of Chaos in an Imitation Game in *Artificial Life III*, edited by C. G. Langton, Addison-Wesley,CA (1994)
3. J. P. Crutchfield, *Physica D* (in press)
4. E. S. Yoshikawa and T. Ikegami in this proceedings
5. T. Hashimoto and T. Ikegami (preprint)