Chapter 2

Evolution of Grammar Systems

2.1 Introduction

2.1.1 Language as an Evolutionary System in Ensemble Structures

Linguistic expressions look quite complex but look far from random. Therefore we feel that there exists a linguistic code. It is said that a grammar of a given language decides a structure of expressions. A grammar constitutes a part of the code. It is determined in part by the community which uses it. It is also commonly assumed that one has to have internal knowledge of one's language when one can derive and recognize appropriately structured expressions. Internal knowledge varies from person to person. We henceforth will refer to individual internal knowledge as an individual grammar. An individual grammar can undergo changes under physical and cultural environment and in its interaction with other grammars.

Language reflects historical factors such as evolutionary paths and growing processes, as shown in the next subsection. Therefore language can be treated as an evolving system. For these reasons, we have to discuss how a grammar evolves in an ensemble of individual grammars and how its complexity evolves.

It is often assumed that the complexity of language is mere reflection of the complexity in the world we live in, just as the complexity of living systems is said to be the reflection of their complex environments. But language is a highly autonomous system, with its own evolutionary dynamics.

MacLennan [72] has studied communication among simple rule agents. Agents exchange information about the local environments in which they live by emitting signals to each other. Fixed signals serve as names of objects which an agent encounters in its environment. Different names may correspond to the same object, resulting in the emergence of synonymous. Complex naming of objects reflects the complexity of the external environment. In Werner and Dyer's model [103], diversity of language is attributed to spatial inhomogeneity of the environment where agents live.

However, we believe that even without complex space or information, grammars can evolve and diversify by intrinsic mechanism. In general, evolving systems, such as evolutionary games [71, 50], Tierra world [84, 85], network of tapes and machines [52, 51] which is studied in the next chapter, constitute notable examples of evolutionary mechanisms. Each example has its own diversifying mechanisms such as host-parasite dynamics and self-referential paradox. If linguistic expressions are not mere labels of external objects, communication helps to develop grammar structures [1].

We use our language not just to describe entities and states of affairs in the external world but also to express our own thoughts. Language must be complex enough to express complex thoughts or concepts. On the other hand, language is used not only in order to communicate with others (exo-language), but also to construct one's "internal models" (endo-language). We regard grammar systems as representations of internal models. Exo-language also reconstructs internal models of both speakers and listeners. Language influences thought and vice versa; complex thoughts make language complex and complex language is conducive to the formulation of complex thoughts. To make false statements gives impetus to complex language: speaker must produce complicated expressions or encipher sentences to make it difficult for others to understand them. Listeners must recognize such complex expressions to correctly receive information. Communication therefore can be thought of as a source of rich grammar structures.

To study languages in evolutionary and network context, we consider a simple language game. Each player has his own grammar and communicates with each other by sending sentences. Player takes turns to speak and listen to the sentences. Each player's advantage is counted by what kind of sentences he speaks, how fast he recognizes sentences of the other and how one's sentences are recognized by other players. A player who get lower scores has to change their grammar and the grammars of players with higher scores are likely to be hereditary.

According to N. Chomsky [16], the computational ability of symbolic grammar is categorized into four classes:

type 0 phrase structure grammar

type 1 context sensitive grammar

type 2 context free grammar

type 3 regular grammar.

The higher a grammar is ranked in the hierarchy, the larger a set of words it can generate. To take a simple example, a word set,

$$\{0^n 1^n | n \geq 1\}, \tag{2.1}$$

can not be derived by a regular grammar (type 3) but can be derived by a context free grammar (type 2) and by any grammar higher in the hierarchy. Here a symbol xy is a concatenation

---

[1] Since our model has no external environments, naming of external objects lies outside the scope of this paper. But we will show that the diversification of words and grammars still occur.

of symbols x and y and x^n represents n times concatenation of symbol x. Hence a set (2.1) includes all strings beginning with any non-zero number of 0s followed by an equal number of 1s.

Evolution of grammar will be discussed as a process of escalation in the hierarchy. From a practical point of view, we have to take finite lengths of words and finite derivation steps into account. If we deal only with a finite set of words, e.g. $\{0^n 1^n | N \geq n \geq 1\}$, Chomsky's hierarchical relationship does not always hold. In computation theory, there are no upper bounds to how much time can be put into deriving words and no ensemble structure is considered at all. We need to figure out what kind of grammar has a practical ability to derive and accept words in finite time steps. The computational ability of a symbolic grammar and hierarchy should be studied within an ensemble. Chomsky [17] said that regular grammars are not suitable for modeling the grammars of natural languages. We assume that the primitive structure of a grammar is a regular grammar and will see how it develop into higher grammars.

In the next subsection, we will review examples of evolution of complexity in linguistic systems.

2.1.2 Evolution of Linguistic Complexity

Evolution of language would be divided to two problems. One is the emergence of language, the other is evolution of linguistic complexity. The former is too complicated problem. To understand it we must investigate many subjects as well as linguistics. By the fact that the evolution of brain especially asymmetry of hemisphere is needed to deal with complex and abstract information, we must develop peleoneurology as well as conventional brain science [69, 34]. Since to utter vowels in a loud voice and articulated consonants need developments of vocal tract and facial muscle [69], we have to know the evolutionary path by paleobiological studies. It is said that evolution of stone tools is related with the evolution of language [70]. Teaching complex processes of making tools, e.g. Neanderthal's, seems impossible without language [34,4]. Language represents its own culture. Hence archaeological evidences should be examined. Communication is shown to be specific not only to human society but also to many other living things. There seems to be symbolic behaviors in some animals [64,40].

On the other hand, development of lexical and grammatical complexity after establishment of language provides different but interesting subjects to study. Historical linguistics, child's acquisition of language and pidgin and creole languages are the examples [19].

Pidgin languages are the best examples of inventing new codes induced by mixture of codes. They are new languages when two or more languages fuse [99]. They are not mother tongues of speakers of pidgins. They have their own first language. For example, when superstratum language spoken by European overseers met substratum languages spoken by African slaves on Caribbean plantations, a relatively simpler new language was invented in order to communicate with people from other areas [108, 29].

Romaine reported some examples of simplification, reduction, restructuring and mixture of lexicon and grammar found in the pidginization processes [87]. Bickerton classified grammatical morphemes to two groups with respect to their robustness [10]. Ones are easily reconstructed morphemes if they are lost, e.g. tense, aspect, modality forms, question words, and irrealis complementizer. The others are hardly reconstructed morphemes if they are lost, e.g. gender agreement, number agreement and most prepositions. The former can give a primitive form of grammar. The evolution from grammar with the former type to more complex one including the latter type would have proceeded.

Creole languages are stabilized and extended ones from pidgins. Different from pidgins They are the mother tongues of speakers. Creolization usually proceeds through several generations. A good example is Tok pisin. Creole languages may approach to their superstratum languages after creolization, if the superstratum languages have large effects to the creoles — for example, when the superstratum language is used for education or broadcast. The approaching process from a creole to its superstratum language is referred to as decreolization. Languages between them are called post-creole continuum [99]. Pidginization, creolization and decreolization dynamics is illustrated schematically in Fig. 2.1.

Creolization can occur from any degrees of pidginization [87]. There are languages called "radical creoles". A radical creole is formed in one generation in a society where some pidgin languages are spoken. It is different from any languages previously spoken in the society. An example is Hawaii creole. Bickerton found that radical creoles have similar characters in common even though many of them are geographically separated. For example, they show simple morphologies, often no morphology at all. Three particles expressing anterior, irrealis, progressive used as auxiliary verbs are another example of the common character [9]. It is very suggestive, since we can imagine that language evolves in an invariant fashion. Bickerton argued that when children are exposed only insufficient model for acquisition, an innate syntactic structure arises. He called it "bioprogram" and asserted it was a candidate for the primitive

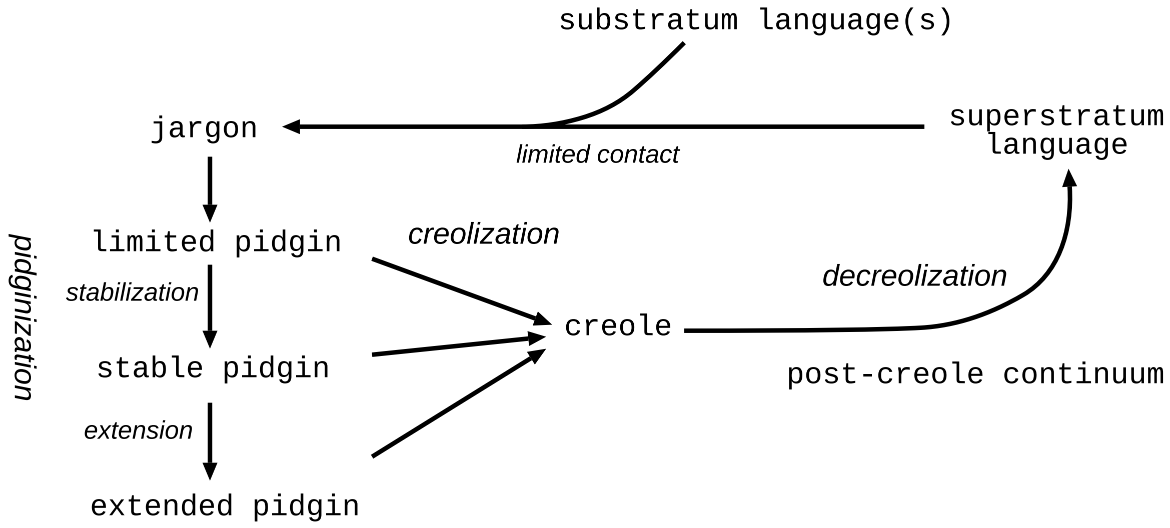

Figure 2.1: pidginization, creolization and decreolization processes

syntax at the emergence of language.

Creoles share some characters with children's language. For example, children often use verbs without inflection. Another example is the fact that they often do not exchange auxiliary verb and the subject word in an interrogative sentence – e.g. “Where I can put it?”. Such sentence is normal in most creoles. Distinction between state and process is another example. In creoles, the progressive particle is by no means used with process verbs as “love” or “like”. Children acquire this distinction very early. They scarcely say as “I am liking her”.

Let us see development of children's language.

At first, children can utter only a word, which is called one word stage. They can recognize adult's syntactic structure, although they cannot speak syntactic structured sentences. It can be said that they have a passive grammar system [80]. At the two word stage, there is no unstressed morpheme as prepositions or auxiliary verbs [1]. Sentences in this stage called “telegraphic speech” [12]. Two main words are extracted from adult's sentences and are combined maintaining their word order in the adult's sentences.

Thereafter, children begin to use some morphemes. Brown reported learning process of 14 grammatical morphemes in English [12]. Present progressive, plural and locative prepositions are mastered in early stages of development. Contractible copula and contractible auxiliary are, on the other hand, acquired at last stages. Although the order is not held always in other language, acquisition order of several morphemes suggests a principle: simple rules are used at first and then complex rules are used in rough ways. To understand this acquisition order, several notions of complexity are proposed – semantic complexity (e.g. [13]), derivational complexity (e.g. [13,32,15]), formal complexity (e.g. [94]), and conceptual complexity (e.g. [18]) (see also Andersen's paper [1]).

Overregularizations have been reported in the course of acquisition of language by cross-linguistic studies [27,93,8,7]. Children typically acquire inflections through the following steps: at first, no inflection; then infrequent and appropriate use of both regular and irregular inflections, e.g. walk-walked, go-went; next, overregularized form – irregular words are inflected as regular words, e.g. "goed" for a past tense of "go"; at last, completely correct use. Pinker interpreted these facts as follows [79]. Children, at first, do not catch a rule of inflection. They only memorize words. Afterword they learn the rule for regular inflections and wrongly apply to irregular words. Thereafter, they acquire whole inflection rules. These facts show that children can generalize underlying rules from finite input and extend from simple to complex rules.

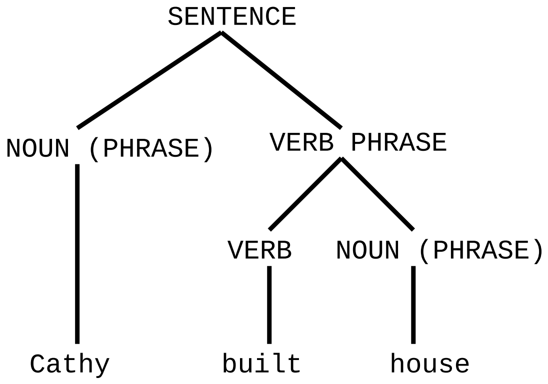

Figure 2.2: A hierarchical structure of a child's sentence

The next stage of development is the emergence of hierarchical structure of sentences. When children speak a sentence, they often say a part of the sentence at first, e.g. they speak a verb phrase, then speak the whole sentence. For example, they say "built house" immediately before speaking "Cathy built house." This fact suggests that their utterances are constructed as in Fig. 2.2. Beside, there is a fact that they never breathe in noun phrase. They seem to regard a noun phrase as a unit. It also suggests that their sentences are hierarchically structured.

In this subsection, we have looked that there are facts showing evolution of linguistic complexity through developments of pidgin and creole languages and children's language acquisition processes. They exhibit conspicuous evidences of dynamical codes. These dynamics are partly induced by mixture of codes – superstratum and substratum languages in the pidginization, and adults' and children's languages in the language acquisition processes.

2.1.3 Organization of this Chapter

Sections in this chapter are organized as follows. Theory of formal language and the Chomsky hierarchy are introduced in §2.2. A model of a language game and evolutionary dynamics are defined in §2.3. Detailed analysis of evolution of those grammar systems is presented in §2.4. Developing an ensemble sharing several words, we call it an ensemble with a common set of words (ECW), is found to be crucial for maintaining diversity of grammar structures. After establishment of an ensemble structure, each grammar system is likely to become complex again. An ECW is discussed in §2.5. A simulation of the model exhibiting a stepwise evolution over several eras will be reported in §2.6. Some other observations will be reported in §2.7. Several implications of ECW and evolution of grammar systems will be discussed in §2.8.

2.2 Basic of Formal Languages

2.2.1 Word and Language

A finite non-void set of symbols is called *alphabet* and is denoted by V. The finite strings of symbols are called *words* over V. The set of all words over V is denoted by V^*. The *length* of a word w is the number of symbols of w and is denoted by $|w|$. An arbitrary set of words is called *language* and is denoted by L.

Any language L is associated with several generative grammar systems G, which is characterized by a set of rules and symbols. By preparing relevant rules and symbols, at least one generative grammar can generate the whole words belonging to the given language L [46].

2.2.2 Generative Grammar

A *generative grammar* G is an ordered four-tuple (V_N, V_T, F, S). Symbols V_T and V_N are disjoint finite alphabet, called *terminal* and *nonterminal* symbols, respectively. A symbol $S \in V_N$ is an *initial symbol.* And a symbol F is finite set of ordered pair (α, β). Here, α and β is word over $(V_N \cup V_T)^*$ and α contains at least one symbol from V_N. The elements (α, β) in F are called *rewriting rules* and will be written in the form

$$\alpha \rightarrow \beta \ . \tag{2.2}$$

2.2.3 Derivation and Acceptance

Rewriting rules are used to derive new words from given ones. If the left-hand of a rule is equal to a part of a word, the part is replaced by the right-hand of that rule.

Given a grammar $G = (V_N, V_T, F, S)$ and two words $X, Y \in (V_N \cup V_T)^*$, we say that Y *is derivable from* X *in one step* and we denote it

$$X \Rightarrow Y, \tag{2.3}$$

iff there are words P_1 and P_2 in $(V_N \cup V_T)^*$ and a rewriting rule $\alpha \rightarrow \beta$ in F such that $X = P_1 \alpha P_2$ and $Y = P_1 \beta P_2$.

Given a grammar $G = (V_N, V_T, F, S)$ and two words $X, Y \in (V_N \cup V_T)^*$, we say that Y *is derivable from* X and we denote it

$$X \overset{*}{\Rightarrow} Y, \tag{2.4}$$

iff $X = Y$ or there is some word $X_0, X_1, X_2, \cdots, X_k (k \geq 0)$ in $(V_N \cup V_T)^*$ and $X_0 = X, X_k = Y$ and X_{i+1} is derivable from X_i in one step $(0 \leq i \leq k-1)$, i.e.

$$X = X_0 \Rightarrow X_1 \Rightarrow \cdots X_{k-1} \Rightarrow X_k = Y. \tag{2.5}$$

When no nonterminal symbols are left in the derived word, a derivation process terminates. If a word X is derived by a grammar G, we say that a word X is *acceptable* by G.

2.2.4 The Chomsky Hierarchy of Grammars

We can classify the generative grammars with respect to a set of rewriting rules. The classification below has been introduced by N. Chomsky [16].

A generative grammar $G = (V_N, V_T, F, S)$ is said to be of *type i* if it satisfies the corresponding restrictions in this list [86]:

$i = 0$ No restrictions, called *phrase structure grammar.*

$i = 1$ Every rewriting rule in F has form $Q_1AQ_2 \rightarrow Q_1PQ_2$, with Q_1, Q_2 and P in $(V_n \cup V_T)^*$, $A \in V_N$, and $P \neq \lambda$, except possibly for the rule $S \rightarrow \lambda$, which may occur in F, in which case S does not occur on the right-hand sides of the rules. Where λ is an empty word which contains no symbols. This grammar types are called *context sensitive grammar.*

$i = 2$ Every rule in F has a form $A \rightarrow P$, where $A \in V_N$ and $P \in (V_N \cup V_T)^*$. This type grammars are called *context free grammar.*

$i = 3$ Every rule in F has a form either $A \rightarrow PB$ or $A \rightarrow P$, where $A, B \in V_N$ and $P \in V_T^*$. This type of grammar is called *regular grammar.*

A language is said to be of *type i* if it is generated by a *type i* grammar. The classes of each type languages are related by the inclusions as follows:

$$\text{regular} \subset \text{context free} \subset \text{context sensitive} \subset \text{phrase structure}. \tag{2.6}$$

2.3 Modeling

2.3.1 Agent with Generative grammar

We define a communicative agent with generative grammar as follows:

$$G_i = (\{S, A, B\}, \{0, 1\}, F_i, S) \ . \tag{2.7}$$

All agents have the same sets of nonterminal and terminal symbols, and are identified by index i. A symbol F_i is a set of rewriting rules peculiar to each agent. The rewriting rules are written in the form $\alpha \rightarrow \beta$ as above mentioned. Here, the left-hand of any rule, denoted by α, is a symbol over V_N. The type of grammar which an agent can have is either a context free grammar or regular grammar. The right-hand of rule, denoted by β, is a finite string of symbols over $V_N \cup V_T$. The same symbol should not be included in α and $beta$ to forbid a self loop. In this paper, we use 0s and 1s for alphabets so that words become bit strings.

2.3.2 Communication

Agents communicate with each other by trying speaking and recognizing words in terms of its own grammar.

All agents derive words using their own rewriting rules. To derive a word a leftmost symbol equal to left-hand side of a rewriting rule is rewritten by a right-hand of the rule. Derivation always starts from an initial symbol S. If there are more than two fitting rules in an agent's rule set, the agent selects one rule randomly. When no nonterminal symbols are left in the derived word, a derivation process terminates. And the derived word is spoken to all agents. An agent fails to speak a word when (i) the derivation does not terminate within 60 rewriting steps or (ii) there is no applicable rule in its rule set. The maximum length of a word is M here. The words longer than M are truncated after the M-th symbol and then are spoken to. The possible number of words (N_{all}) is limited to $2^{M+1} - 2$, and a full set of words speakable by an agent G_i is denoted by $L_{\mathrm{sp}}(G_i)$.

For example, an agent which rules are $S \rightarrow A, A \rightarrow B$ and $A \rightarrow 01$ can derive only $\{01\}$. This agent always rewrites the initial symbol S to A via a rule $S \rightarrow A$. Then there are two rules to be applied, $A \rightarrow B$ and $A \rightarrow 01$. One rule is selected randomly from these two rules. If $A \rightarrow 01$ is selected the word become 01 which doesn't consist of any nonterminal symbols and are not rewritten any further. Then he speaks the word "01" to all agents. On the other hand, if the rule $A \rightarrow B$ is adopted, the derived word is B. Since there is no rule to rewrite, this agent fails.

Agents try to recognize words by applying their own rule in the opposite direction. If an agent can rewrite a given word back to the symbol S within 500 rewriting steps, we say that the agent can recognize the word. Detailed algorithm for recognition process are explained in Appendix. The language recognized by a agent G_i is denoted $L_{\mathrm{rec}}(G_i)$. Note that the inclusion relationship (i.e. $L_{\mathrm{sp}}(G_i) \supseteq L_{\mathrm{rec}}(G_i)$) holds, because of the truncation and limitation of rewriting steps.

In the above example, the agent can understand only a word "01" by writing back it to the symbol S as

$$01 \Rightarrow A \Rightarrow S \ . \tag{2.8}$$

It cannot recognize any other word since the agent does not have applicable rules as *non-terminal symbol* → *terminal symbols* except for $A \rightarrow 01$.

2.3.3 Language Game

We set a language game in a network consisting of P agents. Each agent takes turns to speak a word and it is given to all the agents. Then every agent including the speaker tries to recognize it. Each time step consists of R rounds. For each round, every agent has an opportunity to speak.

Each agent is ranked by three different scores:

1. **speaking** How long and how rare words an agent speaks.
2. **recognizing** How long words and how quickly an agent recognizes.
3. **being recognized** How long words which an agent speaks are recognized.

A word spoken by the l-th agent to the m-th agent at a round c is denoted by the symbol $w_{lm}(c)$. The scores for l-th agent at a round c is computed as follows:

For computing a score of speaking, it is given by,

$$p_l^{\mathrm{sp}}(c) = \begin{cases} \frac{|w_{lm}(c)|}{trend+1} \ , & \text{for speaking a word } w_{lm}(c) \\ -\frac{M}{2} \ , & \text{for failing to speak any word ,} \end{cases} \tag{2.9}$$

Where *trend* is defined as the frequency of the word spoken in the last 10 time steps. An agent gets a higher value of $p_l^{\mathrm{sp}}(c)$ when it speaks longer words and/or less frequent words.

For computing a score of recognizing, it is given by,

$$p_{kl}^{\mathrm{rec}}(c) = \begin{cases} \frac{|w_{kl}(c)|}{s} \ , & \text{for recognizing a word spoken by the } k\text{-th agent} \\ & \text{in } s \text{ rewriting steps} \\ -|w_{kl}(c)| \ , & \text{for not recognizing a word spoken by the } k\text{-th agent} \ . \end{cases} \tag{2.10}$$

A quick recognition of a long word provides a higher value of $p_{kl}^{\mathrm{rec}}(c)$.

For computing being recognized score, it is given by,

$$p_{lm}^{\mathrm{br}}(c) = \begin{cases} \frac{|w_{lm}(c)|}{P} \ , & \text{if the spoken word is recognized by the } m\text{-th agent} \\ -\frac{|w_{lm}(c)|}{P} \ , & \text{if the spoken word isn't recognized by the } m\text{-th agent} \ . \end{cases} \tag{2.11}$$

Agents recognizing each other makes a value of p_{kl}^{br} high.

The total score for the l-th agent in a time step is an average of a weighted sum of three scores over R rounds:

$$p_l^{\mathrm{tot}} = \frac{1}{R} \sum_{c=1}^{R} (r_{\mathrm{sp}} p_l^{\mathrm{sp}}(c) + r_{\mathrm{rec}} \sum_{k=1}^{P} p_{kl}^{\mathrm{rec}}(c) + r_{\mathrm{br}} \sum_{m=1}^{P} p_{lm}^{\mathrm{br}}(c)) \ , \tag{2.12}$$

where r_{sp}, r_{rec} and r_{br} are the respective weighting coefficients. For example, if r_{br} is given a positive value, those who can be recognized by more agents get scores more. But if the value is given negative, being recognized is no more favorable.

2.3.4 Evolutionary Dynamics

In each time step, new agents are produced. They inherit a rule set from their parents with a slight change. The change of rule set is defined by the following three processes:

adding mutation A new rule is added, which is transfered from parents by modifying randomly selected rule.

replacing mutation A randomly selected rule is replaced with a modified rule.

deleting mutation A randomly selected rule is deleted.

The modification ways are following;

1. Replacing a symbol of the left-hand of the rule with the other nonterminal symbol.

2. Replacing a symbol in the right-hand of the rule with the other nonterminal or terminal symbol.

3. Inserting a symbol in the right-hand side of the rule.
4. Deleting a symbol from the right-hand of the rule.

Adding mutation is applied to agents within the rate m_{add}, if their scores exceed the average score. Replacing and deleting mutations are applied to all the agents within the rate of m_{rep} and m_{del}, respectively. The same number of the least scored agents as new agents is removed from a network so that the total number of agents is kept constant.

2.4 Algorithmic Evolution

In this paper, a network consists of 10 agents ($P = 10$) and each agent tries to speak 10 times in each time step ($R = 10$). The score of language game is computed under the fixed parameters: $r_{\rm sp} = 3.0, r_{\rm rec} = 1.0$ and $r_{\rm br} = -2.0$. Note that agents which can speak less acceptable words are benefited for a negative value of $r_{\rm br}$. It is expected that a variety of the words spoken in a population will increase. All the mutation rates are set at equal value 0.04 ($m_{\rm add} = m_{\rm rep} = m_{\rm del} = 0.04$). The maximum length of a word is limited to 6 ($M = 6$), therefore the number of possible words $N_{\rm all}$ is 126.

Initially, all agents assumed to have the simplest grammar, i.e. a single rule with one symbol in the both hand side. They are classified as type 3 grammars due to Chomsky's classification. At least, either a rule $S{\rightarrow}0$ or a rule $S{\rightarrow}1$ should be included in order to derive a word.

We find that evolution of grammar system is accelerated by the characteristic factors, one is a module-type evolution and the other one is a loop forming evolution. Computational ability of agents is measured by the ratio of recognizable words to the total number of possible words, i.e.

$$\text{computational ability} = \frac{N(L_{\rm rec}(G_i))}{N_{\rm all}} \ , \qquad (2.13)$$

where $N(L_{\rm rec}(G_i))$ is the number of words which can be recognized by the agent G_i. Fig. 2.3 represents the example of evolution of the computational ability from the initial network. The computational ability, as well as the number of the distinct words spoken in the network, we call a *variety* of words, evolves in the course time.

A tree that displays the derivation path of a given word is called a derivation tree of the word. We put all possible derivation tree of a grammar system in a directed, connected graph. A structure of the graph expresses the algorithm of the grammar. Algorithmic evolution can be seen in the topological changes of this graph.

2.4.1 Evolution during the Early Stage

It is shown in Fig. 2.3 that computational abilities of agents slowly evolve during initial 200 time steps. In Fig. 2.4 (a)∼(c) the corresponding grammar systems are depicted in graph diagram. The initial agent has a weakest ability, having a direct derivation rule $S{\rightarrow}0$ (Fig. 2.4(a)). The agent can increase the ability by the process of the adding mutation. Adding the rule $S{\rightarrow}1$ to the initial graph generates a branch structure (Fig. 2.4(b)). Further, the multi branch structure

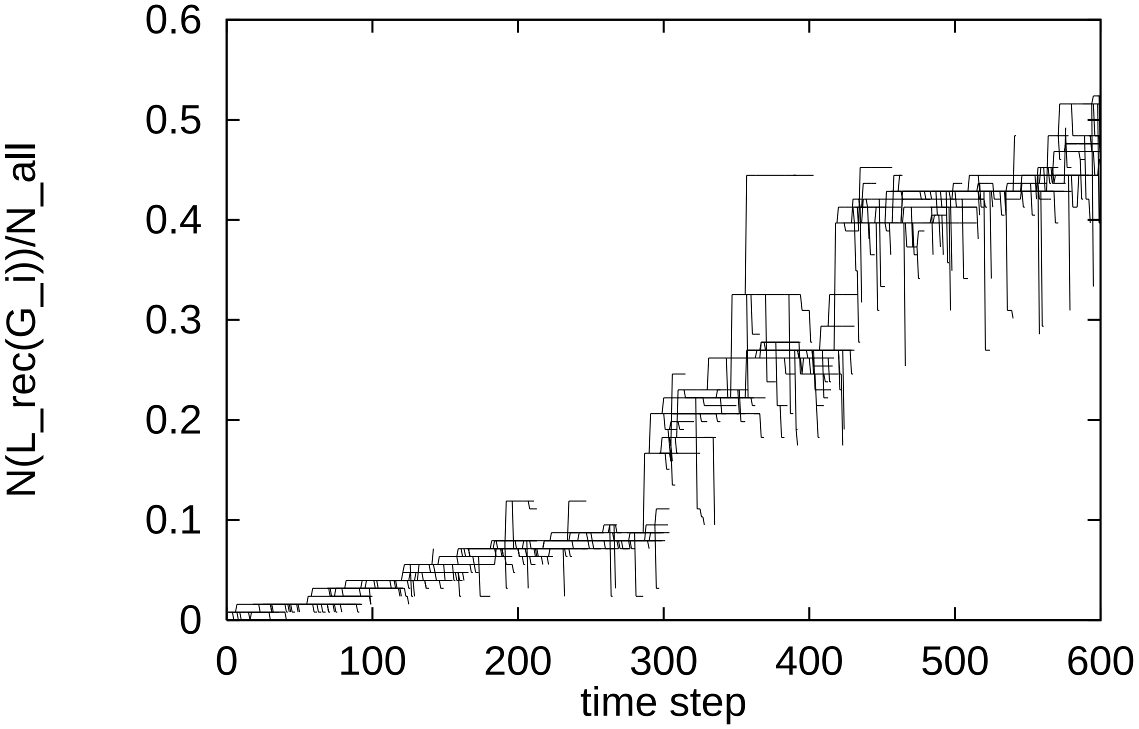

Figure 2.3: Time step v.s. $N(L_{\mathrm{rec}}(G_i))/N_{\mathrm{all}}$. Each line connects one agent to oneself or its offsprings. It branches off by the mutations. A line terminates when the corresponding agent is removed. These lines show upward trend. In initial 200 time step, computational ability gradually increases. After that, transitions to higher computational agent are frequently observed.

will evolve (Fig. 2.4(c)).

2.4.2 Module-Type Evolution

We find in Fig. 2.3 that an agent with the remarkably high ability (> 0.1) appears at time step 192. The change of grammar at this time step is sketched in Fig. 2.4(d). An acquired rule $A \rightarrow 00$ can double the size of acceptable words. Every intermediate word containing a symbol A can be rewritten by the rule $A \rightarrow 00$. In the sense that one common rule is used by many different words to make new words, we call the key rule a module rule. Evolutionary processes driven by the emergent module rule are called module-type evolutions.

2.4.3 Emergence of Loop Structure

Grammar systems can evolve by an alternative evolutionary process, that is, loop structures are formed in a grammar system. Fig. 2.3 shows that a new agent with more powerful grammar appears the population around time step 310. The new agent has a loop structure in its grammar system (see Fig. 2.4(e)). A loop structure can derive a potentially infinite numbers of words recursively. A grammar with a loop structure is categorized as a type 2 grammar or higher one in Chomsky's hierarchy.

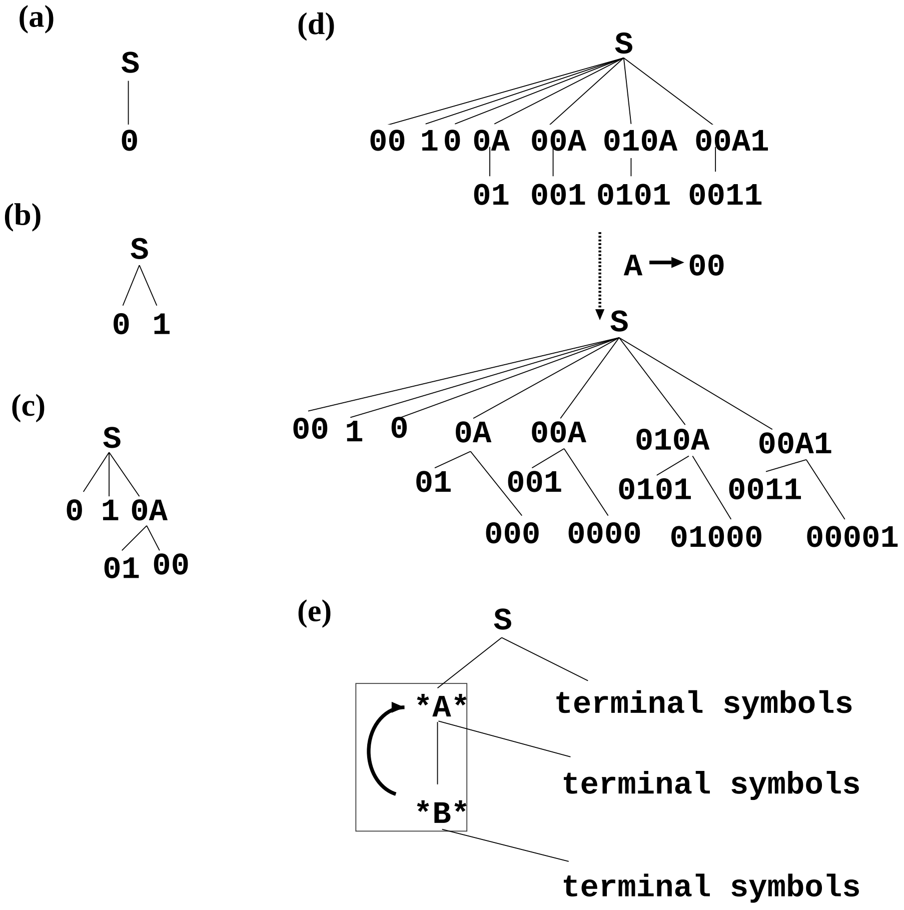

Figure 2.4: The examples of grammar structure are shown by graph diagrams: (a) a sequential structure, (b) a branch structure and (c) a multi branch structure. In the (d) an example of module-type evolution is shown. Acquiring a rule $A \rightarrow 00$, a grammar without bifurcation (upper tree) is evolved into one with bifurcated branches (lower tree). An example of grammar having a loop structure is schematized in (e). Asterisk stands for any symbols. With this grammar, new agent can rewrite words $*A*$ into $*B*$ and vise versa. Words derived from such grammar can not represented in a tree form.

2.5 Ensemble with a Common Set of Words

An upper structure, which is named an *ensemble with a common set of words (ECW)*, emerges in the population of agents. An ECW consists of agents which can speak and recognize a common set of words. The other agents which can't speak or recognize the common set of words are less benefited than those belonging to the ECW.

When there exists an ECW, even an agent of a high ability in a population will die out. For example, a new agent evolved by a module-type evolution dies out at time steps 192 and 310 in Fig. 2.3. Agents will be removed from the network nevertheless they have a power grammar.

At time step 403 an agent with the highest computational ability in the population is dies out (see Fig. 2.3). Agents taking too much rewriting steps to recognize very frequent words are likely to decrease their fitness. We indicate this fact by Table 2.1. The rewriting steps needed to recognize the well spoken words are shown in this table.

Agents which cannot recognize frequent words in the population will be removed in order. An agent G_{306} (agent with ID 306) , which has the second highest ability in the population, is removed first at time step 400. An agent G_{302} which cannot recognize a word "10" is next agent to be removed. At the next time an agent G_{307} which cannot recognize a word "0001" is removed. An agent G_{276} which has the highest computational ability in the population is removed at time step 403,as it cannot recognize the word "00". To stay in the ensemble, where a word "00" is the most commonly spoken, each agent should speak and recognize the word quickly. An ability to speak a kind of words quickly should be balanced with an ability to speak many but long words.

Numbers in bold font in Table. 2.1 represent first two larger rewriting steps to understand the words in the leftmost column. It is clear from this table that it takes much more time for agents G_{306} and G_{276} to recognize several words. To take more rewriting steps to recognize commonly spoken words of the majority is disadvantageous for the agents G_{306} and G_{276}. If a group containing agents G_{306} and G_{276} constituted the majority, the words as "001011" or "010101" would be the commonly spoken words. In such cases, agents G_{306} and G_{276} will take advantageous.

Fig. 2.5 shows the phylogeny of agents existing at time step 400. It shows that the group consists of the agents G_{276} and G_{306} and that of the other agents forms the different lines. They form two different ECWs. The agents in the major ECW have lower computational ability than those in the minor ECW consisting of G_{276} and G_{306}. Two ECWs conflict to survive in

Table 2.1: This table shows rewriting steps to recognize some words (the left most column) spoken at time step 400. Simulation parameters are $r_{\mathrm{sp}} = 3.0, r_{\mathrm{rec}} = 1.0 and r_{\mathrm{br}} = -2.0$. In the second column, the trend of each word (frequency of words in the last 10 time steps) is written. Numerals in the first row are ID of each agent at time step 400 in order that the earlier a agent is removed the lefter it is located. If the agent can't recognize the word, no numerals is indicated. Bold numerals represent the first two longest steps to recognize the left most word.

word	trend	**306**	302	307	**276**	305	301	299	294	290	284
001001	30			121	**185**			121	**145**	133	121
011001	20			157	**423**			166	**214**	190	157
11100	16			52	**245**			46	**58**	52	52
11	14		**7**		**12**	**7**	**7**				
110	12		13		**32**	**14**	**14**				
00110	26		53	**57**		**62**	48	43	54	49	53
110010	11		**174**	147				121	**202**	160	147
00	69	**3**	**3**	**3**		**3**	**3**	**3**	**3**	**3**	**3**
10	57	**7**		5	**7**	5	5		5	5	5
011010	24	**431**	210	164	**431**	120	180	166	251	209	164
001010	31	**233**	134	124	**236**	106	106	116	161	141	124
1110	24	**125**	45	27	**122**	60	60	24	29	27	27
01110	9	**122**	91	69	**192**	55	76	66	83	75	69
00101	25	**91**	65	58	**91**	52	49	57	66	63	58
111	30	**53**	21	13	**52**	24	24	13	13	13	13
0001	78	**20**	19		**20**	21	16	18	18	18	17
001011	14	**401**			**404**						
010101	10	**190**			**193**						

the network. Those in the major ECW behave cooperatively as the result by speaking and recognizing a common set of words and get higher scores. At last all agent in the minor ECW is removed from the network. In this way the evolution toward the high computational ability is suppressed by establishing ECWs.

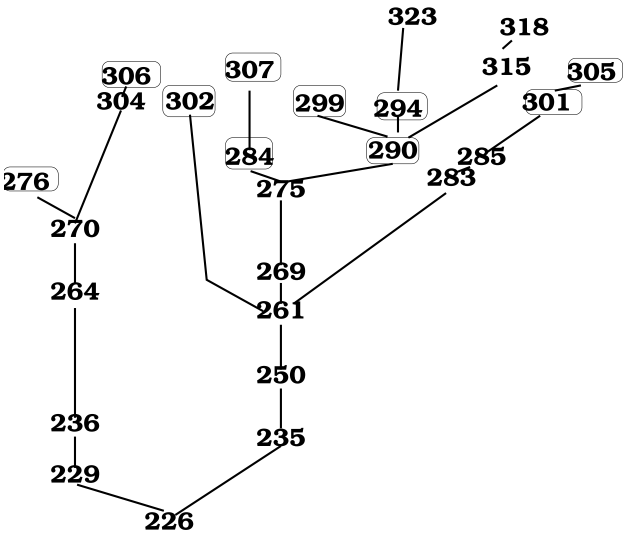

Figure 2.5: This picture represents the phylogeny of agents at time step 400 (in oval boxes) from common ancestor (G_{226}). A number represents ID of each agent. A line is drawn from parent agent (lower) to its offsprings (upper). Two genetic series are bifurcated from the common root (G_{226}), the agents G_{306} and G_{276} and that of the other agents. They are forming different ECW. The agent G_{306} and G_{276} are both contained in the left series.

Fig. 2.6 displays a situation of spreading out an ensemble in network. A high trend peak moves from low number of agents which recognize some words to high number of agents with time step going on. The broken arrow in this figure indicates this movement. It implies that new agents belonging ECWs come into existence and at last the ECWs dominate whole network.

After removing agents G_{306} and G_{276} from a network, an ECW spread to whole network as shown in Fig. 2.6. Then agents come to compete with each other within the same ECW. Proportional to the number of rewriting steps to recognize the commonly spoken words, the agents are removed from the network. In the ECW, a new agent with the high computational ability will then emerge through algorithmic evolution.

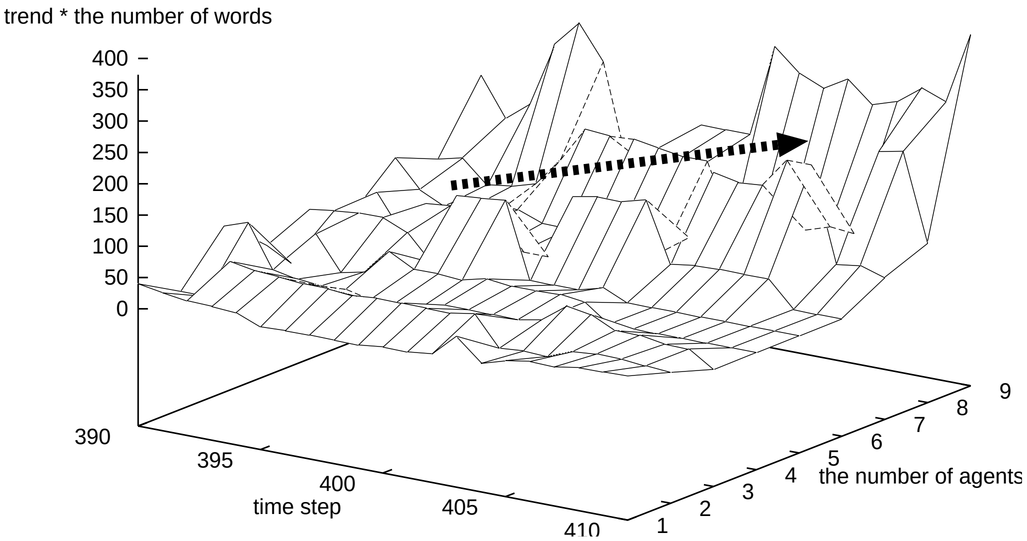

Figure 2.6: time step v.s. the number of agents which can recognize some words v.s. trend *times* the number of the words which each number of agents can recognize : We can see that an ECW spread out in network. This graph has a peak at time step 396 and the number of agents 5, where some words which 5 agents can recognize are spoken many times. After that this peak moves to higher number of agents in the course of time. That is to say, agents speaking the words increase. Increase of trend of particular words seems reasonable to say that agents in some ensembles increase. At last all of agents can speak and recognize such words. Namely, some ensembles have dominated whole network. The broken arrow indicates the movement of the peak.

2.6 Punctuated Equilibrium

We have seen that our system shows rapid algorithmic evolution in the grammar systems in certain stages. Since, algorithmic evolution is suppressed by forming ECW, rapid algorithmic evolution follows the quasi-equilibrium stages. Temporal evolution of amounts of handling information therefore shows punctuated equilibrium phenomena (Fig. 2.7). Handling information defined below is sensitive to the formation of ECW.

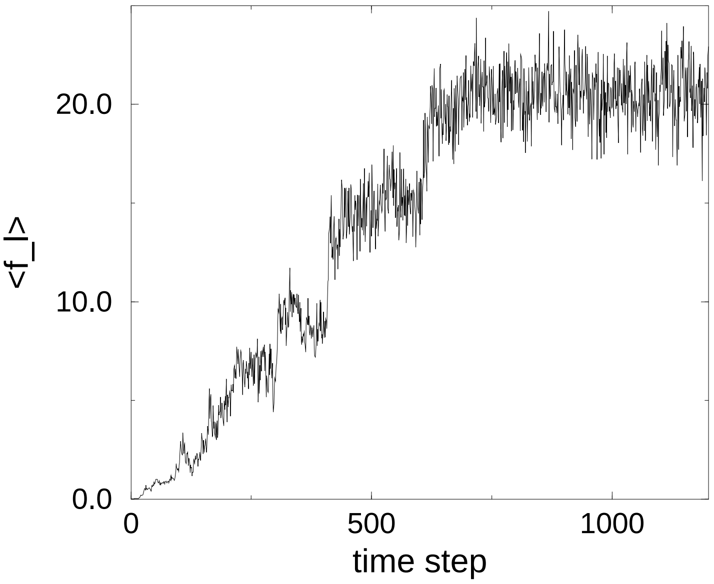

Figure 2.7: time step v.s. the average handling information (see the definition in the text): In the first 700 time steps, stepwise evolution can be observed. The stepwise changes reflect alternate evolution of ECW and the algorithmic evolution.

The handling information of the l-th agent is defined as the follows,

$$f_l = \frac{1}{RP^2} \sum_{c=1}^{R} \sum_{k=1}^{P} |w_{kl}^{\mathrm{rec}}(c-1)| \sum_{m=1}^{P} |w_{lm}^{\mathrm{rec}}(c)| \tag{2.14}$$

$$|w_{ij}^{\mathrm{rec}}(x)| = \begin{cases} 1 & \text{if } x = 0 \\ \text{the length of } w_{ij}(x) & \text{if } x > 0 \text{ and the word which spoken by the} \\ & \quad i\text{-th agent is recognized by the } j\text{-th agent} \\ 0 & \text{otherwise .} \end{cases} \tag{2.15}$$

Information contents of a word is simply given by the length of a word. The initial amount of handling information, i.e. $|w_{ij}(0)|$, is defined as 1.

If an agent gets high value of f_l, which suggests that the agent can recognize words spoken by others and its speaking words are recognized by other agents. When some ECWs conflict with the other ECWs, the averaged handling information in a population,

$$\langle f_l \rangle = \sum_{p=1}^{P} \frac{f_l}{P} \ , \tag{2.16}$$

does not increase. After some ECWs occupy the whole network, long and new words will be spoken and recognized again by agents. Punctuated equilibrium phenomena in the amount of $\langle f_l \rangle$ is explained by the scenario.

2.7 Results of Other Simulations

2.7.1 Minimal Almighty

We can make a *minimal almighty* agent. It is an agent which can speak and recognize all possible words with the least number of rules. For example, a minimal almighty agent has the rules such as:

$$S \rightarrow A, A \rightarrow SS, S \rightarrow 0, S \rightarrow 1 \ . \tag{2.17}$$

This grammar is categorized as a type 2 grammar. It recognizes and speaks all the words very quickly. However, it tends to speak a low variety of words because of random adoption from plural fitting rules. A minimal almighty agent cannot invade into the system composed of ECW because it can speak lower variety of words. In the case of $r_{\mathrm{rec}} = 1.0, r_{\mathrm{sp}} = r_{\mathrm{br}} = 0.0$, an almighty agent can evolve, since low variety of speaking words has no disadvantage for its fitness. An example of evolution of computational ability when an almighty agent emerges is shown in Fig. 2.8. Mutants not only from the almighty agent but also ancestors of almighty agent have much lower computational ability than its parent as shown in this figure. The fragility of the almighty's grammar suggests that the rules in almighty agent strongly interact with each other.

2.7.2 Score of Being Recognized

We have three parameters in the definition of total score (see eq. 2.12). Here, we will see the effect of the parameter r_{br}. If it is given positive value, a tendency to recognize each other will be encouraged. But if it is given negative, the opposite tendency begins to spread. We display, how it depends on the parameter r_{br}. If r_{br} is larger than 4.0, both the variety of word spoken and the average handling information are suppressed. Since agents gets higher score by speaking and recognizing the same and short words. If r_{br} becomes less than -3.0, the average handling information will go down (Fig. 2.9 b)) since it reflects the degree of being recognized. Variety, on the other hand, is kept middle level in the range $r_{\mathrm{br}} < -3.0$ (Fig. 2.9 a)). In such range, agents make higher score by speaking less frequent and less recognizable words. It maintains variety of words in the middle level even when r_{br} is less than -3.0. In earlier stages (time step < 1500), simulation with $r_{\mathrm{br}} = -2.0$ has shown quick evolution. However it is saturated at $r_{\mathrm{br}} = 0.0$ and below (time step ≥ 1500). We conclude that a slight negative value of r_{br} accelerates the evolution at the highest speed.

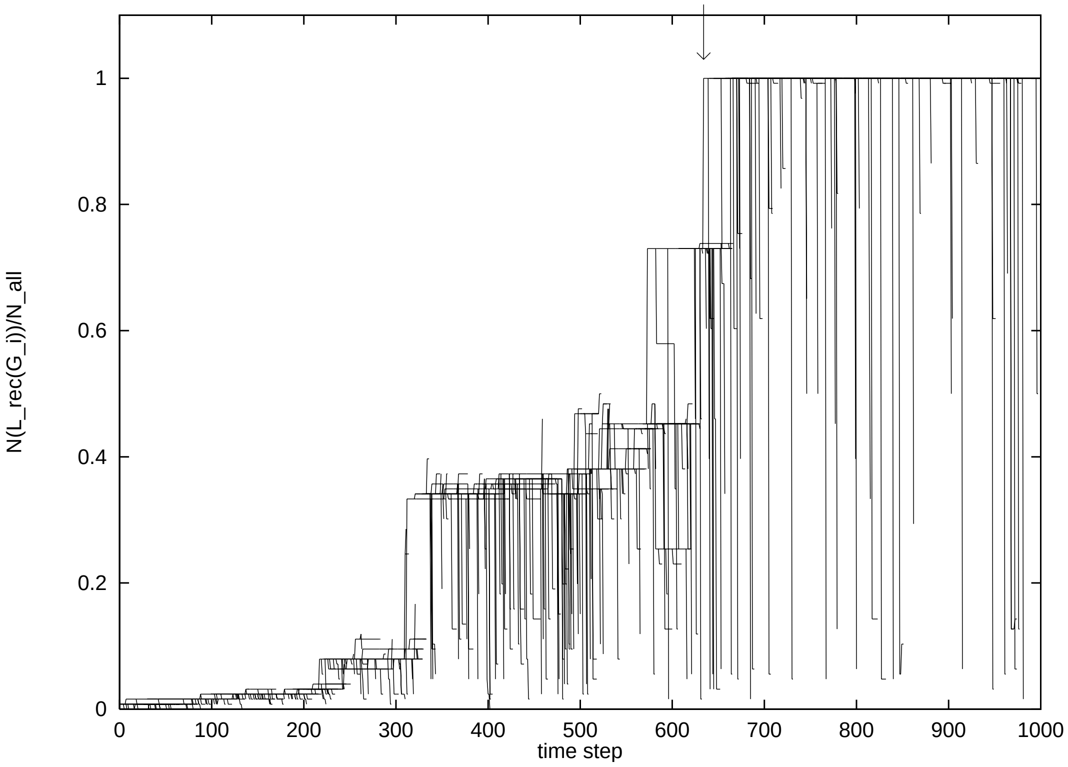

Figure 2.8: An example of emergence of almighty agent. Time step v.s. $N(L_{\mathrm{rec}}(G_i))/N_{\mathrm{all}}$. Each line connects one agent to oneself or its offsprings. It branches off by the mutations. A line terminates when the corresponding agent is removed. Coefficient for each score is $r_{\mathrm{rec}} = 1.0, r_{\mathrm{sp}} = r_{\mathrm{br}} = 0.0$. An almighty agent emerges at time step = 634 (the arrow indicates). Almost mutants from both almighty agent and almighty's ancestor have very low computational ability.

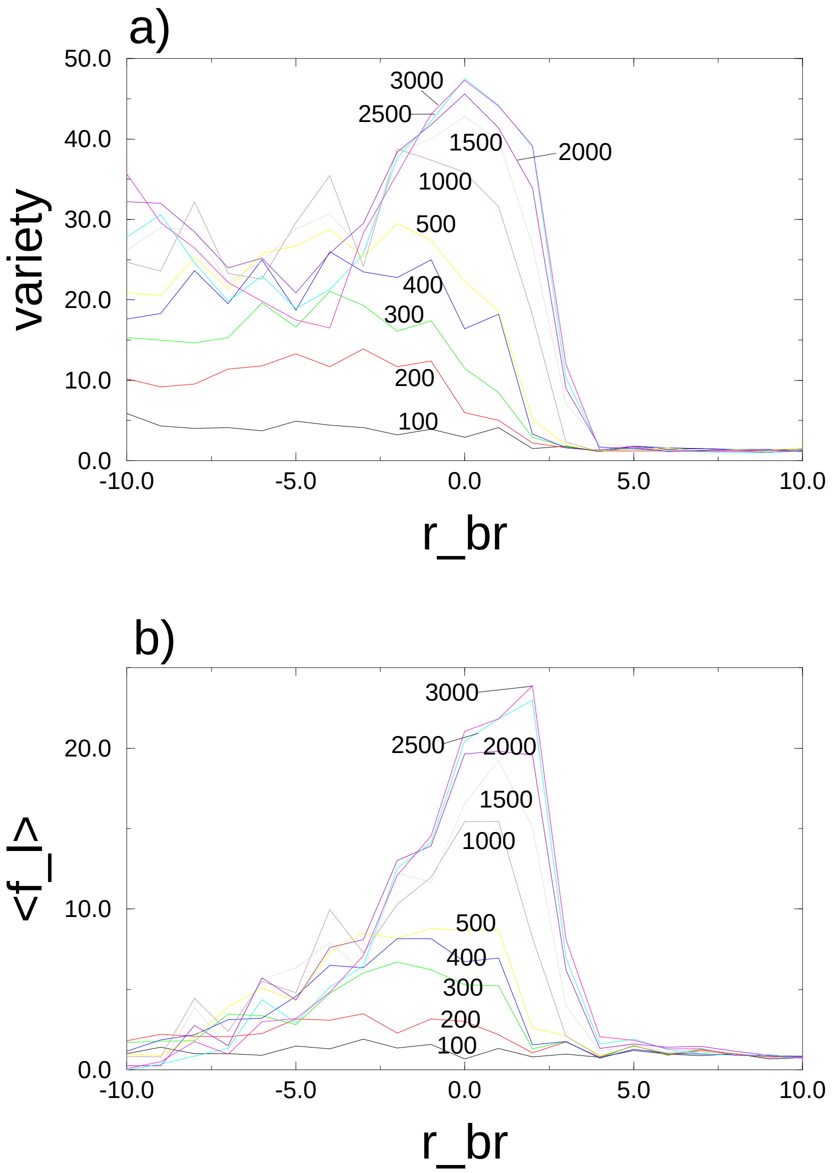

Figure 2.9: a)variety of words spoken and b)the average handling information vs $r_{\rm br}$, coefficient of score of being recognized, at several time steps. The other parameter values are $r_{sp} = 3.0$ and $r_{\rm rec} = 1.0$. The number attached to each line represents time step. a) In lower range of the parameter ($r_{\rm br} \leq -3.0$) variety shows middle level, in midway of the parameter ($-3.0 \leq r_{\rm br} \leq 2.0$) it has high variety and in the higher range it shows low level. b) For each range $< f_l >$ shows low, high and quite low level, respectively.

2.8 Discussions

In the present study, we have shown two types of evolutionary dynamics. One is a module-type evolution. What it means for a rule to be a module is that it can be utilized by other rules in a grammar to generate nearly twice as many words as before. The number of recognizable word rapidly increases when this module emerges in a grammar. The other type is a loop forming evolution. A grammar equipped with a loop structure can derive recursively many words. A grammar with a loop structure cannot be represented in a tree shape. Thus the grammar system climbs up Chomsky's hierarchy from type 3 to type 2 by acquiring loops. Hence we call such a loop forming evolution an algorithmic evolution. We consider two significant evolutionary processes in natural language as possible correspondents of our evolutionary dynamics.

In natural language, there is a process called affixation, whereby a group of letters is added to a word to produce another word. Affixes added to the heads of words are called prefixes; un-, mis-, pre- are examples in English. Ones added to the ends of words are called suffixes; -ful, -less, -ish are such examples. The module-type evolution we have found in our simulation can be related with word formation processes involving affixes. The module rules are used to produce new words by attaching themselves to other words. They indeed behave as affixes.

The other kind of process produces nested sentences. Complex sentences, phrases within phrases and clauses within clauses are such examples. For example, "A man with a colorful umbrella who is walking over there will be a candidate of the political party which suffered a setback." This sentence can be decomposed into four independent simple sentences without any nesting: "A man has a colorful umbrella. He is walking over there. He will be a candidate of a political party. The party suffered a setback." This nested structure is a very important feature of natural language to produce rich and complex sentences. If natural language were only required to serves as a mere signaling tool, complex syntactic structures might not have evolved. In order to produce nested sentences, a complex grammar is likely to appear.

An ECW (ensemble with a common set of words) is characterized by a shared set of words. Individual grammars comprising an ECW are restricted to deriving words freely. A particular usage of words in an ECW emerges. Thus a code is organized. In order to speak and recognize sharing words quickly, it is advantageous to have their words as single rules (i.e. $S \rightarrow$ words), and to combine several rules of nonterminal symbols on their right hand sides (e.g. $A \rightarrow 0BS$) to speak/recognize other words. This can be regarded as double articulation in natural language. Because a sentence consists of words and a word consists of symbols, we can have infinite sentences with finite symbols. The double articulation also constitutes one of the most important features of natural language. Such structure is good for recognizing frequently spoken words quickly and speaking many longer words. When we only count a recognizing score in a language game, almighty agents appear and a variety of words is suppressed, since such structure as double articulation is not likely to develop.

In biological systems ranging from ants' society to human society, social laws and norms develop. Essential problem in a development of norms is how such ordered upper structures evolve from local interactions [60, 98]. It is suggested that homeochaos or the edge of chaos is formed to establish cooperation [58, 110]. In the iterated prisoner's dilemma game, the Tit For Tat strategy is formed in cooperative societies [2]. Once such a social structure appears, it in turn restricts local dynamics. A recursive loop between global structures and local structures continues endlessly. Taylor described such a recursive relationship as "LOCAL to GLOBAL back to LOCAL, inter-level feedback loops" [98]. This is stressed as evidence of "emergence", which is one of the central concerns in artificial life studies [98, 63].

We here propose "children's play" as an example of such emergent phenomena. Children sometimes change rules of a game while playing it. The rules dictate how the player is to behave and how new rules emerge from children's play successively. Such dynamics of children's play is like the dynamics of linguistic code. It is reported from sociolinguistics studies (e.g. [92]) that in general linguistic communities, social dialects are frequently observed. They are language variations which are used only in particular social classes or in different groups based on various social variables, such as social classes, education levels, occupations and age groups. Peculiar usages emerge from conversations in a small group. Once they are established, they constrain the variety of words. It is said that group words such as social dialects are likely to emerge when a group consists of about 8 members [92].

Within our language game, an ECW produces dialects, irrespective of individual grammars. We have used a term *net-grammar* to refer to the way an ECW produces dialects and restricts the potential computational ability of individual grammars in the proceeding paper [41]. We can say from our study that a code has restrictive force to individuals sharing the code. Linguistic codes can be a source of social codes.

Our language game favors the following agents: speak long and infrequent words; recognize long words quickly; speak words which are not recognized by other agents. These conditions force players to try to produce more expressions. As a result, diversity of words and complexity of grammar get enhanced. Such a tendency of players is valid through the whole situation, a tendency to speak/recognize a common set of words emerges in an ECW. Agents trying to speak the same words is tantamount to trying to communicate with each other. This is what we believe to be one of the main purposes of language. Conflicts between enhancing expressivity and establishing linguistic code result in punctuated equilibria in communicating information.

Agents in our game are in a dilemma. They try to speak words which others cannot recognize, but at the same time, to recognize what others say based on the same grammar. This dilemma somehow resembles imitation games proposed by Suzuki and Kaneko [96, 59]. It was a simple model for mutually mimicking birds. In their papers a bird with a complex song was stronger when it came to defending its territory, since complex songs were difficult for other birds to imitate. But at the same time, a bird imitated other birds' song by the same mechanism. A song is a time series generated by a logistic map. Evolution of a bird's song was achieved by mutating the parameter of logistic map. They observed evolution of a song towards the edge between the periodic window and chaos, that is, birds singing songs of marginal stability seemed stronger.

Crutchfield has shown that complexity of automata which has accepted time serieses of dynamical mappings has been the highest at the onset of chaos [23, 22]. In his model, input to automata was quantized time serieses of maps. By replacing this input by output of other automata, there may be a similarity to our model. Ensemble of Crutchfield's automata may give rise to complexity by language game.

Algorithmic evolution and restriction by an ECW causes punctuated equilibria even in language systems. It is brought about by two factors, modification of grammars and network structures. Punctuated equilibrium in genetic systems has been produced by genetic fusion operators, which combine genes with module genes [53]. (Note that the module rules which we define in the present study are different from the module species used by Ikegami and Kaneko [53]. The module species were frequently utilized as fusion partners by *many other* species. The module rules, in contrast, are used to speak or recognize many words within *a single* agent.) By crossover operators different schemata can evolve in parallel, being combined into a better fitness [45]. Consequently, intermittent patterns will appear if crossover is introduced as genetic operator. There are no module agent or parallel evolutions in the present study, as we do not incorporate such operators as fusion or crossover that combine one's grammar with others'. But rules in a grammar have epistasis, that is, they interact with each other to produce words, a slight modification of a rule may cause large changes in the ability to speak and recognize words and thus the algorithmic evolution can occur.

If agents given randomly generated words, the variety of words produced according to grammars exhibits no punctuated equilibrium, it almost linearly increases, because agents cannot form an ensemble structure.