Chapter 4

Conclusions and Developments

We define a *code* in a system as ways of using symbols and interpretations of information. We refer to successive emergence and dynamical changes of codes as *evolution of code*. Evolution of code in a communication system and a genetic system have been studied here.

4.1 Summary and Conclusion — Evolution of Grammar Systems

In Chapter 2 we have studied evolution of communicating agents in a network. To discuss evolution of linguistic code and grammar systems, we proposed a language game. Agents communicate with each other by deriving and accepting words in terms of their own generative grammar. They are ranked according to their communicative effectiveness. Agents which speak either long infrequent words, recognize long words quickly or speak words which are not recognized by the other agents overpower others. Mutation of rules are introduced to update agents' grammar. A code in this system is characterized by particular usage of words. Evolution of grammar systems was discussed as evolution of computational ability in the Chomsky hierarchy.

Computational ability of each agent's increases as well as variety of words spoken in the course of time. We have found that information in communication shows a stepwise evolution, where equilibrium states and sudden jumps occur alternately.

In the equilibrium states, agents communicate with common words. We call the community exchanging a common set of words an ECW. Different ECW is characterized by a different common words. When different ECWs conflict, an ECW composed of higher computational ability agents does not always outwit. An ECW with more number of agents wins. That is, ECWs suppress for individual grammar to evolve. Code of communication is discussed in terms of ECW. A particular usage of words determined within an ECW is a code.

On the other hand, breaking up ECWs with evolving grammar is caused by mainly two remarkable mechanisms. One is a module-type evolution. An agent can enlarge a set of recognizable words by acquiring a module rule. The role of modules may be corresponding to that of affixes in natural language. The other mechanism is a loop forming evolution. A grammar containing a loop structure can recursively derive words. If a grammar forms a loop, it can potentially derive infinite number of words. Such grammar can derive nested sentences. Long and rich expressions are produced by phrases within phrases or clauses within clauses structures. A tree to loop structural change presents an evolution from regular to context-free grammar.

The emergence of code and evolution of computational ability can also be seen by different parameter settings. But for the extreme case, the results are very different. Variety of words and information in communication remains very low, if ability for being recognized is a big advantage. In this case, only a code is established but it does not exhibit dynamics. In the opposite limit, agents speak many long words but without mutual understandings. In other words, any code is organized in the network.

Let us mention different points from related works. MacLennan [72] and Oliphant [77] think that codes in communication is owing to matching between each symbol and its referent among agents. But we think that the meaning and indication must not be thought to be identical (cf. [33, 82]). We assume that codes are based on usages of phrases. Therefore evolution of diversity of words and complexity of grammars are brought about without diverse external world different from MacLennan's model [1]. Werner and Dyer [103] has discussed that diversity of language attributes to physical barriers. Since our model consists of only 10 agents because we are interested in dynamics of formation of group word or trendy word in relatively small group, we cannot discuss diversity of code. However, we indicate that conflict between two codes prevents evolution of grammar systems and it results in punctuated equilibria evolution.

4.2 Summary and Conclusion — Coevolution of Tapes and Machines

In Chapter 3, coevolution of tapes and machines are studied. A machine reads a tape and then a new tape and a new machine are produced. We have simulated an ensemble of machines and tapes to discuss computational aspects of mutations and translations. By introducing external noise, we have investigated the role of randomness on machine's deterministic action. In this model, a tape can be interpreted as distinct machines depending on machines reading the tape. Changing tape symbols by machines' deterministic actions is named "active mutation". Probabilistic changing under external noise is called "passive mutation". A code of replicating network is, therefore, a whole set of interpretations of tapes by machines composing the network.

We see the evolution from a minimal self-replicating network to a larger one by increasing external noise. If external noise is small, the minimal network is a final state of evolution. A minimal network consists of a stable pair of machine and tape, which are reproduced locally. When external noise excesses a certain threshold, a spontaneous transition to a large stable net-

[1] Our further research suggests that the diversification of words and complication of grammars can be allowed without preference for long and rare words in the definition of score function.

work is observed. After the transition, the network retains a self-maintaining network involving stable core structure, named a “core network”. Core networks replicate individual tapes and machines globally.

Dynamical states of the populations of tapes and machines in core networks can be fixed point or oscillating state. Large core networks have high active mutations. They seem to mimic the reaction paths which external noise have forced on.

Stability of core networks is attributed to the combinations of embedded autocatalytic loops. A fixed core consists of several autocatalytic loops. In oscillatory cores, the number and topology of embedded autocatalytic loops changes in the course of time. Autocatalytic loops can be divided into two type. One is named as Eigen-Schuster (ES) type, the other double loop type. Each tape in an autocatalytic loop of ES type is locally replicated. On the other hand, machines and tapes in that of the other type are globally replicated. The latter network has non-zero active mutation. Large autocatalytic networks with inhomogeneous couplings are stabilized by the latter one.

Interpretations of tapes in a network can not be determined locally. It must be organize consistently in the whole network. In a fixed core network, a set of interpretations are fixed. Thus a code for self-reproduction is organized. A network chooses one possible interpretations system to replicate the structure of network. Global restrictions for interpretations make a self-consistent logic in a network. If a logic does not made consistently, we call it *logical openness*. In an oscillatory core network, a code for self-reproduction is not stable. Oscillation of code and network topology can be related to logical openness. A set of interpretations are destabilized by itself. We refer to such self-destabilizing nature of code as *openness of code* .

At last we give a brief summary of new feature in the present study. Our model is development to ensemble context from Neumann's self-reproducing automaton [102]. And autocatalytic network is studied from evolutionary point of view different from Eigen and Schuster's work [24]. Indeed, we show evolution from minimal loop to large and complex autocatalytic network. It is revealed that introducing active mutation process stabilizes complex networks prevailing Niesert's catastrophes [76]. Moreover, the autocatalytic network in our study can have inhomogeneous couplings. Oscillation of network topology is a novel feature.

4.3 Further Developments

We are interested in oscillation induced by logical self-denial structure. A system with a code is threatened the maintenance of the code by itself. The underlying mechanisms of oscillation of network topology in our machines and tapes system are not clarified yet. We can say that oscillation of network topology can be an extension of logic oscillation, which is first discussed in Brownian algebra [11]. Such oscillating nature of logic can naturally occur in our model.

We exemplify other interesting subjects to discuss from the viewpoint of evolution of *code*.

- Social code

  Norms, common senses or ethics in a society govern behaviors and ways of thinking of members in the society. Most individuals have to obey them without intentions. They are spontaneously organized in a society, regulating behaviors of the constituents of societies. We refer such regulations in a society as social codes. Social codes cannot be determined in advance. Evolution of social code is also induced by internal dynamics and logical openness.

- Self-reference code

  A self-referential paradox of laws are worth noting. There are laws to legislate the way to enact laws. We find a self-referential paradox, when we make laws for laws. Hofstadter has introduced an interesting self-modifying game named NOMIC [43]. Players of the game make or modify rules of the game. Players can make some rules which forbid to make or modify other rules. This is an example of logical openness, since laws themselves destabilizes the laws.

- Code in the immune system

  The immune system distinguishes self and non-self in our body. Codes in the immune system are regulations to discriminate self image from non-self image. But what is self or non-self cannot be determined in advance. It is distinguished by whether the immune system responds to antigens or not. A network model of the immune system [55] characterizes the system as the network of antibodies stimulating each other. Both internal and external antigens' images are restructured by the network. The self/non-self image is not dynamically stable. An antibody which constitutes the self image may become non-self. The whole system may be destroyed by some internal/external antigen images. Selecting a network structure may be considered as constructing a code of self/non-self. Therefore a notion of dynamical code developed in our study must be compared with the immune network.

- Code for articulation

  We at first perceive our continuous external world, then articulate it. Assigning symbols comes after the articulation. Codes for articulation determine how articulate our continuous external world and how assign symbols. However, the dualism – the external world is continuous and the internal world is discrete – can be criticized (e.g. [49]). And we may not symbolically process information in our internal world. Lakoff insists that a conceptual system is metaphorically structured and emphasizes the importance of physical experiences [67, 66]. We think that the code for articulation can be determined partially through communication among agents and it can vary temporally and locally (cf. [90, 104]). Articulating the external world and structuring internal world is remained as future problems.

4.4 Remarks on Artificial Life Studies

Langton has proposed Artificial Life studies in his manifesting paper [68]. He summarized the essential features of computer-based Artificial Life models as follows:

- They consist of populations of simple programs or specifications.
- There is no single program that directs all of the other programs.
- Each program details the way in which a simple entity reacts to local situations in its environment, including encounters with other entities.
- There are *no* rules in the system that dictate global behavior.
- Any behavior at levels higher than the individual programs is therefore emergent.

Since both models we have reported here satisfy these features, our studies are kinds of Artificial Life Studies.

Langton and other many researchers on Artificial Life say that "emergence" is the key concept in the field [68, 63]. Here the concept means that the appearance of global or higher level structure which cannot be expected from local or lower level dynamics and interactions [3]. Our main results, ECWs and core networks, are such emergent global structures. We showed not only the appearance of global structures but also dynamical changes of the global structures, which is our main concern – *emergence and dynamics of code*. Taylar stressed the feedback from higher level to lower level as he described "LOCAL to GLOBAL back to LOCAL inter-level feedback loops" [98]. Dynamics, such as successive formation, collapse and oscillation, of global structures are induced from such inter-level interactions.

Computation is another key term in Artificial Life studies. Some classical works has been devoted to treat biological systems or language as computational systems. Neumann's theory of self-reproducing automaton has been a pioneering work of computational treatment of biological system [102]. Chaitin has tried to make an information theoretical definition of life in terms of algorithmic complexity theory for descriptions of biological systems [14]. Chomsky has opened the way to consider language as a computational process with derivational grammar theory [17].

Emmeche has divided concept of computation into four classes as [25]:

- The formal, or algorithmic, concept of computation, which has its theoretical footing in the notion of Universal Turing Machine.
- An informal, intuitive, or "mathematical" concept of computation that is not bounded by the known limitations of formal systems.
- A biological concept of computation. This seems to be a quasi-theoretical concept that can be understood in many ways: for example, as problem solving by learning and adaptability; as molecular processing of information in cells; or as computation by neural networks.
- A physical concept of computation, that might be nonrepresentationalistic. The entities that cooperate in computational enterprises are patterns that can transmit, store, and modify information, but these patterns seemingly do not have to "stand for" anything, as long as no functional constraints are imposed from a higher level.

Our studies are relevant to an informal or a biological concept of computation. We have modeled each agent in our models with simple information processing unit – with derivational grammar systems in the study of linguistic code, and binary strings and Turing-Machine-like automata in the study of genetic code – as the first feature of Artificial Life models listed above. By this modeling we try to catch computational aspects of dynamics of language, self-reproduction and catalytic networks. We has showed in the former study that gaining a loop structure, we named it algorithmic evolution, has enriched the diversity of recognizable words. We has indicated in the latter study that active computation as active mutation has permitted large and complex autocatalytic networks which have been able to storage larger information than hypercycle type autocatalytic networks. Unfortunately, the attempts have not been brought completion. More attention should be paid to interrelations among concepts of computation.

In addition to the concept of computation, the important roles of chaos for biological systems and cognitive systems has been pointed out [100, 58, 54, 57]. Our concern is not only relations between the formal and an informal, and the formal and a biological concept of computation in many agents system with dynamical nonlinear interaction [2] but also roles of chaos in the system. In the formal computation and formal logic system, it is known that there is undecidability such as halting problem of Turing Machine or Gödel's incompleteness theorem [37]. Such undecidability is induced by difference between internal and external viewpoint of the system. It is an interesting point to discuss how such undecidability is caused in modeled biological or cognitive system with constructing internal observers. [3] The importance of an internal observer to understand not only cognitive systems but also physical, chemical, biological system has been pointed out by Rösller [88], Conrad [20] and Tsuda [101, 56].

The aim of Artificial Life studies is said to give pictures of *life-as-it-could-be* [68]. Namely, we want to construct *lifelike* phenomena in artificial media such as computers, robots or bio-materials. Whether a phenomenon is lifelike depends on the definition of life. The definition has been discussed for a long time.

However, we cannot give a settlement to the problem. There must be gray-zone whatever criteria for life is stated. There will exist phenomena which we will thought of as alive but not satisfying these criteria, and phenomena which we will not thought of as alive but satisfying these criteria. Anyway, one of the most large contribution of Artificial Life studies is that they raise discussions on what it means to be alive. But, the discussions often fall into realism or externalism. We are likely to think a phenomenon as lifelike if it is displayed with creature-like representation. Life is neither substance nor concrete state, but is a process arises from cognitive

[2] Relation between dynamical system and formal computation system has been studied by Moore. He proved that Turing Machine could be embedded to two dimensional piecewise linear map [74, 75]. His work is remarkable result connecting dynamical systems and computation. It has been practically shown that dynamical systems can have different type undecidability from the sensitivity to initial value usually referred to as chaos.

[3] Undecidability in physical system has been also studied, for example by Svozil [97].

activity. We think that the process always makes unexpected behaviors – truly emergence.

Since such emergent property through cognitive action is essential for evolution, approaching with the viewpoint of internal observer to constructing complex systems comes to be imperative [56]. To model a system with internal observer, not only creatures but also cells or molecules [20,39] should be treated as internal observers. The system consisting of their activities has some kinds of arbitrariness or undecidability, but behavior of the system has to be assigned restrictions by validities peculiar to the system. Cohering between internal and external viewpoint results in dynamics of the system. Such dynamical mechanism will be correlated with the oscillation of code and network topology observed in the study of network of tapes and machines.