Phenomenal Computing Carrying a Weak Paradox as Indefinite Environments

Yukio-Pegio Gunji[a, b] Tatsuji Takahashi[b] & Moto Kamiura[b]
a Department of Earth & Planetary Sciences, Faculty of Science, Kobe University
b Graduate School of Science & Technology, Kobe University
yukio@kobe-u.ac.jp (corresponding author)

Abstract
The brain model is defined as a phenomenal computing consisting of explicit computing subsystem and its environment carrying the execution of computation. By introducing the modified infomorphism (Barwise & Seligman, 1997) as an operator between sub-systems, a model of phenomenal computing is expressed as a weak paradox. Such a model can explain genesis of module in a brain and duality of conscious explicit cognition and subconscious implicit perception in an abstract sense.
Keywords: Infomorphism, Semantics, Abstract brain, Computing, Lattice

1 Introduction

We propose the notion of phenomenal computing as a dynamical pair consisting of a computing system and the environments of executing computation (Gunji et al., 2004a,b; Gunji 2003). In this framework, all phenomena including materialistic systems and neural networks generating consciousness are expressed as a system consisting of lower- and upper-level subsystems. Origin of emergent upper level (e.g., consciousness) can be constructed as the genesis of explicit difference between lower- and upper-level from an initial condition in which upper-level is latent. We show that the metaphorical toy model of phenomenal computing leads to the duality consisting of a robust computing system and instable environments. The idea of phenomenal computing is beyond the simple binary opposition between machinery recursion and emergent property, and can explore the internal perspective for a system within indefinite environments (Gunji, 2004).

Whenever one argues regarding autonomy, emergent property, and/or consciousness, one adheres to examine them in terms of binary opposition such as the computable vs non-computable. In that perspective the notion of computing is latently based on machinery recursion. Imagine that a machine outputs an error because of overheat derived by going on computing in a hot room. Because such a computed error is out of the binary opposition, it is necessary for a machine with latent errors to consider a machine and its environments in a lump, and that induces the idea of phenomenal computing.

International Journal of Computing Anticipatory Systems, Volume 14, 2004
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-00-8

To express phenomenal computing as a toy model, we introduce two important ideas, (i) infomorphism (Barwise & Seligman, 1997) and (ii) a self-reference exposed with the frame problem (Gunji et al., 2004a,b). Infomorphism is defined as a pair of maps (inter-subsystems transformation) between two subsystems (including intra-subsystem transformation), and is developed from situational semantics. Therefore, two subsystems can correspond to a computing system and the definite environments of executing computation. Although an infomorphism is a tool to express computing machine and its environments in a lump, "computing within environments" is destined to be a self-referential form entailing a paradox. The question arises how one uses self-reference against a paradox. The answer is in Kripke's skepticism (Kripke, 1982). A Kripke's skeptic makes a self-reference in invalidating the premise of self-reference. That is why it can be used as a framework by which a self-reference is positively expressed against a contradiction.

A self-reference under the framework of Kripke's skepticism is expressed as the mixture of intra- and inter-subsystems transformations in the infomorphism. It is a model of phenomenal computation consisting of a system and not definite but indefinite environments. We estimate the time development of subsystems in a term of lattice, and show a robust pair of Boolean (system) and non-distributive lattice (environments), it suggests an individualized computing system against instable environments. Especially we focus on difference between non-autistic and autistic children's reaction for Sally-Ann task (Wimmer & Perner, 1983) and explain the difference in terms of phenomenal computing.

2 Phenomenal Computing

In starting from the naïve hypothesis that a brain is a machine and especially that perception of red is computing the stimulus of red, there can be two different logical objections for this hypothesis (e.g., Preston & Bishop, 2002; Chalmers, 1998). The one objection is based on self-reference: If perception of red is replaced with computing red, one obtains a statement such that (perception of) red is computing red. The statement is directly expressed as a self-referential form such that

$$\text{red} = \text{computing (red)} . \qquad (1)$$

In substituting computing (red) for a variable red in the formula, computing(red), one obtains an infinite form such that red = computing (computing ... (red)...), that is a contradiction because of the indeterminacy of value of red. The other objection is based on, what is called, the frame problem (McCarthy & Hayes, 1969; Dreyfus, 1992). The notion of computing red is, in fact, ambiguous. What can one define computing in a term of a working machine? If one defines a working machine by "computing by a machine connected with a convenient outlet", such a definition is demolished when the power station is crushed. It leads to adding power station to the definition of a working machine. It results in infinite regression of the definition such that computing(red) = computing with a machine, power station, oil transportation, ... (2)

As a result, one cannot define computing (red) in a definite finite form. That is a frame problem.

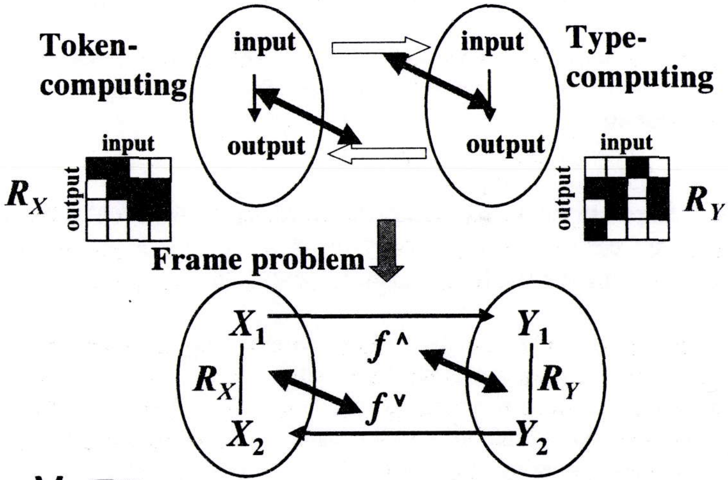
If two objections, self-reference and frame problem are addressed independently, two objections are seriously accepted and there can be no way to overcome the objection. However, if two objections are addressed contemporaneously, one can construct a self-reference so as to avoid a contradiction (Gunji et al., 2004a,b). The premise of self-reference is demolished by frame problem, and vice versa. In the objection based on self-reference, a contradiction results from ambiguous indication of red that is a premise of the objection. The one is a part of a formula, computing (red), and the other is a whole formula. By contrast, the objection based on the frame problem, addresses indefiniteness in indicating a whole of formula, computing (red). Therefore, if one constructs a self-referential form, red = computing (red), under the condition exposed of frame problem, one can obtains a model of perception of red so as to avoid a contradiction.

That is the first step of phenomenal computing. Phenomenal computing consists of a part of explicit computing and its environments to execute computing. Both these two parts are regarded as computing systems. The explicit computing is called type-computing and environmental computing is called token-computing. If one constructs a model of perception in this framework, one is faced with a self-referential form. Even if a neural activity regarding a particular perception is divided into two parts, type- and token-computing, environmental computing is embedded in an explicit computing. As well as a problem of "red = computing (red)", the model is destined to be a self-reference resulting in contradiction. To avoid a contradiction, one has to introduce the property of frame problem in this framework.

The next question arises, what is a formal expression of frame problem. In our context, it is weak premise of self-reference. In the sense of category theory (e.g., Goldblatt, 1991), a self-referential form derived from a diagonal argument in which a functor between two categories is mixed up with an arrow in a category. Because a functor is defined so as to preserve composition of arrows, a self-reference results in a contradiction. Therefore, weakening a functor not to preserve composition is a way to weakening a premise of self-reference and is a formal expression of frame problem. One of ways to weaken a functor is replacing a functor with an infomorphism. That is the second step of phenomenal computing.

First we define a computing system as a triplet; a set of inputs and outputs defined as two sets, and computing operation expressed as a binary relation between them. As mentioned before, phenomenal computing consists of type- and token-computing, and each of them is defined as a triplet. A triplet can coincide with a relation category (i.e., objects are defined as sets, and arrows are defined as relations). Two functors between categories can be defined. If two computing systems are connected by a functor, self-reference results in a contradiction. To avoid such a contradiction, a functor is replaced with an infomorphism as mentioned in the next section.

Phenomenal computing is re-defined as follows: Two computing systems (each of them is expressed as a triplet, two sets and relation) and an infomorphism. In this framework a self-reference is expressed as a mixture of relation and an infomorphism that leads to time development of whole system (Figure 1).



$$\forall x \in X_1 \quad \forall y \in Y_2 \qquad f^{\vee}(y)\, R_X\, x \Leftrightarrow y\, R_Y\, f^{\wedge}(x)$$

Figure 1: Schematic diagram of phenomenal computing. First, consider a system consisting of two computing subsystems, type- and token-computing that are connected with two functors (thick white arrows). If a self-reference that is defined by a mixture of an arrow and a functor (bi-directional arrows) is constructed in this framework, it results in a contradiction. To avoid a contradiction, a functor is replaced with an infomorphism that is a pair of two maps, $f^{\wedge}$ and $f^{\vee}$ satisfying a particular condition.

It is clear to see that a brain is a typical example of phenomenal computing (Tye, 1996; Gunji, 2004). Consider a Brocca and/or Wernicke area concerning about linguistic activities. Even if a linguistic module is regarded as a machine in the sense of approximation, other regions of a brain are also employed to neural activities to execute boundary conditions for a linguistic module (Ramachandran, 1998). Therefore, duality of linguistic module and other regions can correspond to explicit type-computing and its environments (token-computing). Note that a linguistic module appears in the process of development, by self-organizing process. There is no explicit foundation by which a module and its environments are distinguished from each other. We have to explain genesis of duality of explicit computing (i.e., module) and its *implicit environments*.

Especially we focus on the difference between non-autistic children's and autistic children's reaction for Sally-Ann task (Wimmer & Perner, 1983) (also see autism and savant syndrome in Sacks, 1985; Rimland & Fein, 1988; Sullivan, 1992; Selfe, 1977; Snyder & Mitchell, 1999; Happe, 1996; 1999). Sally-Ann task is expressed as a character play for a subject. The play story proceeds by the following:

(i) Sally places a ball in a basket in a room, and then she goes out.
(ii) Ann enters the room and finds a ball in a basket. She hides a ball in a refrigerator.
(iii) Sally comes back to the room.

After a subject listens to such a play story, he is asked, "Where will Sally search for a ball?" Non-autistic children answer that Sally searches for a ball in a basket. By contrast, many of autistic children answers that Sally searches for a ball in a refrigerator.

Psychologists explain that autistic children cannot imagine others' mind. It is clear to see that our mind is based on finite knowledge. As well as all of us, what Sally knows is just finite, and she never knows that Ann hides a ball in a refrigerator. As a result, Sally searches for a ball in a basket. Therefore, psychologists conclude that autistic children cannot imagine other's mind.

We think that difference between non-autistic and autistic children's reactions is based on the difference in constructing the relationship between parts and a whole. If a subject can distinguish parts from a whole story, he can distinguish the truth of an individual character from the truth of a whole story. Then, he can distinguish Sally's own truth (a ball in a basket) from the truth in a whole story (a ball in a refrigerator). Therefore, it is expected that he answers that Sally searches in a basket. By contrast, if a subject cannot distinguish parts from a whole with respect to logic, he assimilates Sally's own truth with the truth in a whole story. Because the truth in a whole story is unique (a ball in a refrigerator), it is expected that Sally searches for a ball in a refrigerator.

In an abstract sense, it is possible to see that difference between non-autistic and autistic children's decision making is based on difference between logic distinguishing parts from a whole and logic assimilating parts with a whole. In focusing on this topic, we examine behavior of the model of phenomenal computing.

3 Dynamical Infomorphism as an abstract brain

Phenomenal computing consisting of the type/token-computation is formalized by a dynamical infomorphism (Gunji et al., 2004a,b). First, each computation is defined as a classification called by Barwise (Barwise & Seligman, 1997). Firstly, mathematical tools we need are given by following.

Definition 1. Classification (Barwise & Seligman, 1997)

Classification is defined by a triplet $<\mathrm{typ}(A), \mathrm{tok}(A), R_A>$, where typ($A$) and tok($A$)

are sets and R_A is a binary relation between typ(A) and tok(A).

Originally, Barwise regard a classification as a perspective consisting of an objects-set called token and an expressions-set called type. The symbol typ(A) and tok(A) denote the abbreviation of type and token. But, our concepts, type-computation and token-computation have nothing to do with a set of types and a set of tokens. As mentioned beneath, type- (token-) computation is defined as a classification. The relationship between two classifications is defined by an infomorphism. Even if two perspectives are not isomorphic to one another, the communication between them can be implemented by an infomorphism. The duality of the type- and token-computation is expressed as a pair of classifications connected with an infomorphism.

Definition 2. Infomorphism (Barwise & Seligman, 1997).

Given two classifications <typ(A), tok(A), R_A> and <typ(B), tok(B), R_B>, a pair of maps <$f^{\vee}$, $f^{\wedge}$> with $f^{\vee}$:tok(B)$\rightarrow$tok(A) and $f^{\wedge}$:typ(A)$\rightarrow$tok(A) is called an infomorphism if and only if: for all $\alpha\in$typ(A) and all $b\in$tok(B),

$$f^{\vee}(b)\, R_A \alpha \quad \Leftrightarrow \quad b R_B\, f^{\wedge}(\alpha). \tag{3}$$

It is called the fundamental property. It is diagrammatically shown as

$$\begin{array}{ccc} \mathrm{typ}(A) & \xrightarrow{\;f^{\wedge}\;} & \mathrm{typ}(B) \\ R_A \Big| & & \Big| R_B \\ \mathrm{tok}(A) & \xleftarrow{\;f^{\vee}\;} & \mathrm{tok}(B) \end{array} \tag{4}$$

We here call <typ(A), tok(A), R_A> and <typ(B), tok(B), R_B>, token-computation and type-computation, respectively.

A computation as a phenomenon is described as duality of type/token-computation consisting of those two classifications and an infomorphism. Although in Barwise's view a classification is independent of each other, only a pair of classifications makes sense as a computation as a phenomenon in our framework.

The following definition is also available to prove some statements.

Definition 3. Type-set / Token-set (Barwise & Seligman, 1997)

Given a classification <typ(A), tok(A), R_A>, a type-set of a token, a, and a token set of a type, α, is defined by the following respectively:

$$\mathrm{typ}(a) = \{\alpha\in\mathrm{typ}(A) \mid aR_A\alpha\} \tag{5}$$
$$\mathrm{tok}(\alpha) = \{a\in\mathrm{tok}(A) \mid aR_A\alpha\} \tag{6}$$

The fundamental property of an infomorphism can be replaced by the following. From the proposition, given an infomorphism and a classification, the other classification that can constitute the duality of type/token-computation can be determined.

Proposition 1. (Also see (Gunji et al., 2004b)

The condition: $b, b'\in\mathrm{tok}(B)$ with $b\neq b'$, $f^{\vee}(b)=f^{\vee}(b')$ and $f^{\vee}(b)R_A\alpha \Rightarrow bR_B f^{\wedge}(\alpha)$ and $b'R_B f^{\wedge}(\alpha)$; added with the condition: for all $\alpha\in\mathrm{typ}(A)$ and $b\in\mathrm{tok}(B)$, $bR_B f^{\wedge}(\alpha) \Rightarrow f^{\vee}(b)\, R_A\alpha$; is equivalent to that $<f^{\vee}, f^{\wedge}>$ is an infomorphism.

Proof: ($\Rightarrow$): Supposing the mentioned condition, the only thing to be proved is: $f^{\vee}(b)\, R_A\alpha \Rightarrow bR_B f^{\wedge}(\alpha)$. It is assumed that $f^{\vee}(b)R_A\alpha \Rightarrow b\not R_B f^{\wedge}(\alpha)$. If there exists $b'\in\mathrm{tok}(B)$ with $b\neq b'$, $f^{\vee}(b)= f^{\vee}(b')$ and $f^{\vee}(b)R_A\alpha$, from assumption, $b\not R_B f^{\wedge}(\alpha)$, then $f^{\vee}(b')\not R_A\alpha \Leftrightarrow f^{\vee}(b)\not R_A\alpha$. That is a contradiction.
($\Leftarrow$): In supposing $f^{\vee}(b)R_A\alpha \Rightarrow bR_B f^{\wedge}(\alpha)$ and $b'\not R_B f^{\wedge}(\alpha)$, $f^{\vee}(b')\not R_A\alpha$ and $f^{\vee}(b)R_A\alpha$. That is a contradiction.

Proposition 2.

If for all $\alpha\in\mathrm{typ}(A)$, $bR_B f^{\wedge}(\alpha) \Rightarrow f^{\vee}(b)\, R_A\alpha$ and $f^{\vee}$ is mono, .a pair of maps $<f^{\vee}, f^{\wedge}>$ is an infomorphism.

Proof. In supposing that $f^{\vee}$ is mono, α, α $\in\mathrm{typ}(A)$ with $\alpha\neq\alpha \Rightarrow f^{\vee}(\alpha)\neq f^{\vee}(\alpha)$, and then $f^{\vee}(\alpha)=f^{\vee}(\alpha) \Rightarrow f^{\vee}(\alpha)=f^{\vee}(\alpha)$. Therefore, there is no pair $b, b'\in\mathrm{tok}(B)$ with $b\neq b'$, $f^{\vee}(b)= f^{\vee}(b')$. Then from proposition 1, $<f^{\vee}, f^{\wedge}>$ is an infomorphism.

Under those frameworks, the dynamical duality of type/token-computation is defined by the following. Because it is expressed as a mixture of relations and an infomorphism, that is one of formal expressions for weak paradox.

Definition 4. Dynamical Infomorphism

Given four classifications, token-computation ($\mathrm{typ}(A)$, $\mathrm{tok}(A)$, $R_A^{\,t-1}$), ($\mathrm{typ}(A)$, $\mathrm{tok}(A)$, $R_A^{\,t}$) and type-computation, ($\mathrm{typ}(B)$, $\mathrm{tok}(B)$, $R_B^{\,t-1}$), ($\mathrm{typ}(B)$, $\mathrm{tok}(B)$, $R_B^{\,t}$), and two infomorphisms, $<f^{\vee(t-1)}, f^{\wedge(t-1)}>$ and $<f^{\vee(t)}, f^{\wedge(t)}>$, the time transition of relations are defined by the following:

$$aR_A^{\,t+1}\alpha :\Leftrightarrow f^{\vee-1(t-1)}(a)\, R_B^{\,t}\, f^{\wedge(t)}(\alpha) \tag{7}$$
$$bR_B^{\,t+1}\beta :\Leftrightarrow f^{\vee(t-1)}(b)\, R_A^{\,t}\, f^{\wedge-1(t)}(\beta). \tag{8}$$

where f^{-1} is an arbitrary induced map from $f{:}X \rightarrow Y$ such that for all $x \in X$, $f^{-1}f(x) =_f x$, where $=_f$ is an equivalence relation such that $x =_f x' :\Leftrightarrow f(x)=f(x')$. If there exists an infomorphism $<f^{\vee(t+1)}, f^{\wedge(t+1)}>$ such that for all $\alpha \in \mathrm{typ}(A)$ and all $b \in \mathrm{tok}(B)$

$$f^{\vee(t+1)}(b)\, R_A^{\,t+1} \alpha \quad \Leftrightarrow \quad b R_B^{\,t+1} f^{\wedge(t+1)}(\alpha), \tag{9}$$

it is said that the time proceeds.

It is diagrammatically shown as

$$\begin{array}{ccccccccc}
& f^{\wedge(t-1)} & & & f^{\wedge(t)} & & & f^{\wedge(t+1)} & \\
\mathrm{typ}(A) & \longrightarrow & \mathrm{typ}(B) & \mathrm{typ}(A) & \longrightarrow & \mathrm{typ}(B) & \mathrm{typ}(A) & \text{------} & \mathrm{typ}(B) \\
R_A^{\,t-1} \Big| & & \Big| R_B^{\,t-1} & R_A^{\,t} \Big| & & \Big| R_B^{\,t} & \Rightarrow\; R_A^{\,t+1} \Big| & & \Big| R_B^{\,t+1} \\
\mathrm{tok}(A) & \longleftarrow & \mathrm{tok}(B) & \mathrm{tok}(A) & \longleftarrow & \mathrm{tok}(B) & \mathrm{tok}(A) & \text{------} & \mathrm{tok}(B) \\
& f^{\vee(t-1)} & & & f^{\vee(t)} & & & f^{\vee(t+1)} &
\end{array} \tag{10}$$

where $<f^{\vee(t+1)},\ f^{\wedge(t+1)}>$ is searched as an infomorphism.

In general, $aR_A^{\,t+1}\alpha :\Leftrightarrow f^{\vee-1(t-n)}(a) R_B^{\,t} f^{\wedge(t-m)}(\alpha)$ and $bR_B^{\,t+1}\beta :\Leftrightarrow f^{\vee(t-n)}(b) R_A^{\,t} f^{\wedge-1(t-m)}(\beta)$, where m and n are arbitrary natural numbers. The time development defined by definition 4 allows the collapse of the duality of type/token-computation (i.e., there is no infomorphism between the type- and token-computation). Therefore, it is difficult to simulate long-range time development. We propose the approximated model of the time development by the following.

Definition 5. Approximated model of Dynamical Infomorphism

The system of definition 6, of which time can perpetually proceed, is approximately expressed as the following.

(step1)
$$f^{\wedge(t+1)} := f^{\wedge(t)} f^{\wedge-1(t-1)} f^{\wedge(t)}, \tag{11a}$$
$$f^{\vee(t+1)} := f^{\vee(t-1)} \tag{11b}$$

(step2)
$$bR_B^{\,t+1}\beta :\Leftrightarrow f^{\vee(t-1)}(b) R_A^{\,t} f^{\wedge-1(t)}(\beta) \tag{12}$$

(step3) $aR_A^{\,t+1}\alpha$ is defined such that for all $\alpha \in \mathrm{typ}(A)$ and all $b \in \mathrm{tok}(B)$,

$$bR_B^{\,t+1} f^{\wedge(t+1)}(\alpha) \Rightarrow f^{\vee(t+1)}(b)\, R_A^{\,t+1} \alpha. \tag{13}$$

As a result, time always proceeds. Both step2 and step3 can be replaced by the following alternative procedures:

(step2') $\qquad aR_A^{t+1}\alpha :\Leftrightarrow\ f^{\vee -1(t-1)}(a)\ R_B^{\,t}\ f^{\wedge(t)}(\alpha)$ (14)

(step3') $\qquad bR_B^{t+1}\beta$ is defined such that for all $\alpha\in$typ(A) and all $b\in$tok(B),

$$f^{\vee(t+1)}(b)\ R_A^{t+1}\alpha \Rightarrow bR_B^{t+1}\ f^{\wedge(t+1)}(\alpha). \tag{15}$$

With respect to the approximation of dynamics defined by definition 6, the procedures, step2-3 and step 2'-3' are equivalent. Therefore, two procedures can be changed arbitrarily.

Through mixing intra-computing (relation) with inter-computing (infomorphism), both two relations and an infomorphism are perpetually changed. In the context of a brain model, computing concerning about particular cognition (type-computing) and its environmental neural activities (qualia-perception) can be perpetually changed. However, a type- computing is generally robust with respect to logical structure.

We here estimate relation by constructing a lattice or a topological structure. According to definition 2, typ(a) is a subset of tok(A), and then a relation is regarded as a kind of filer by which subsets of tok(A) can be observed. Given a relation, one can construct a lattice by colleting a set of typ(a) for all a in typ(A), and defining a partial order by inclusion, however a lattice cannot be verified.

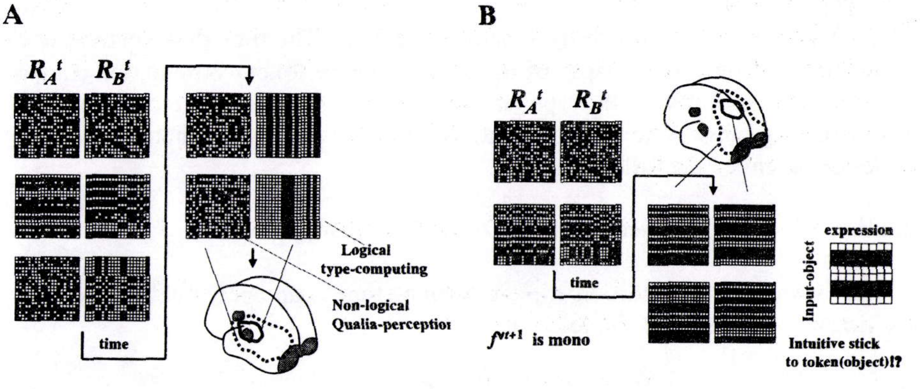


Figure 2: Time development of dynamical infomorphism. A. general time development. B. Specific time development under a condition $f^{\vee}$ is always an injection. See text.

Figure 2 shows a time development of dynamical infomorphism from an initial condition in which both relations of type- and token-computing are randomly chosen. Most of time developments are shown in Figure 2A. Although token-computing is changed in keeping random relation, a relation of type-computing is converged a uniform structure; tok(B) consists of two kind of states β and β such that typ(β) = typ(B) and typ(β) = an empty. Therefore, a lattice representing a relation is defined by {typ(B), an empty}, and that is a Boolean lattice. Figure2B shows a time development in which f^{v} is always constructed as an injection (i.e., mono). In this case, relations of both type- and token-computing are converged into particular relation such that for all β typ(β) is the same as each other. As a result, a lattice is a singleton set of which a part is the same as a whole.

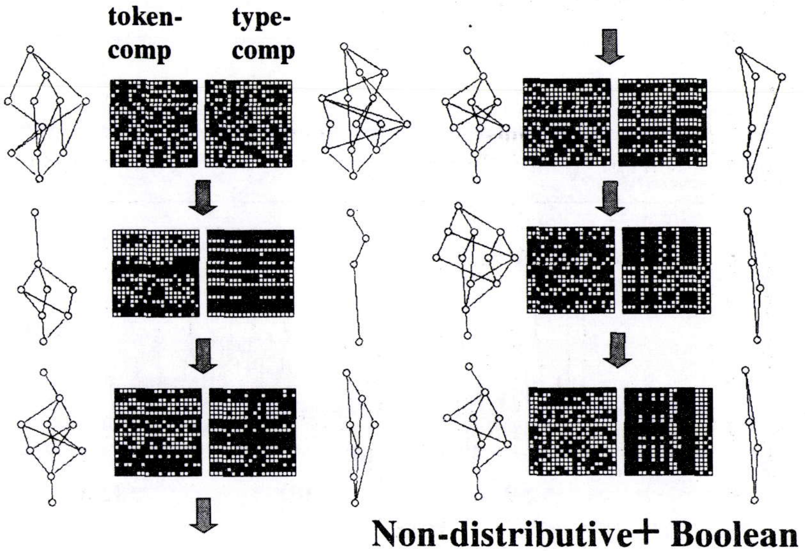


Figure 3: Time development of dynamical infomorphism, accompanied with a Hasse diagram representing a lattice corresponding to a type- and token-relations.

Recall the discussion regarding Sally-Ann tasks. We argue that non-autistic children can grasp the story-universe accompanied with the outside and they distinguish parts from a whole. By contrast, autistic children identify parts with a whole, and then the truth of an individual character is assimilated with a truth of a whole story. These tendencies can be explained by the time development of dynamical infomorphism. It is clear to see that Figure 2A shows a dynamical brain for non-autistic children in which a type-computing appeared in Brocca or Wernicke areas representing Boolean algebra containing parts and a whole and that Figure 2B shows a dynamical brain for autistic children in which both type- and token- computing are similar with each other, representing a singleton set with no distinction of parts and a whole.

Figure 3 shows an example of time developments of the dynamical infomorphism that is more general case. A lattice represented as a Hasse diagram (Figure 3) is defined by a subset of relation. Given a relation, R_A, sub-relation is chosen by Sub(typ(A))$\cap R_A$, where Sub(X) is a subset of a set X. For a sub-relation, one can construct a lattice by colleting a set of typ(a) for all a in typ(A) , empty set and typ(A). In this case, one can always construct a finite lattice. As shown in Figure 3, in general, type-computing is converged into Boolean lattice on one hand, and token-computing is converged into non-distributed lattice.

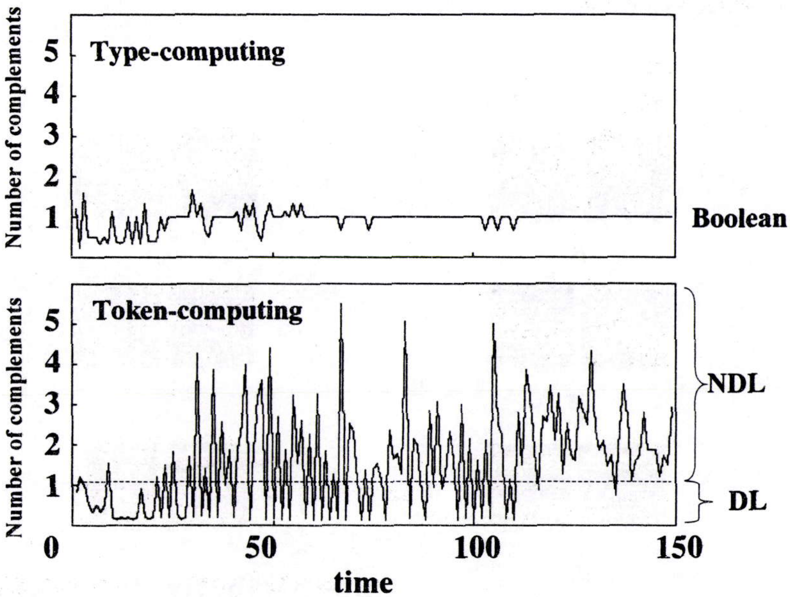


Figure 4: Time development of the mean value of the number of complements for a lattice representing type- and token-relations. NDL and DL represent regions of non-distributive and distributive lattice. Through time development, dynamical infomorphism leads to a dynamical duality consisting of robust Boolean and non-distributive lattices.

To estimate time development of dynamical infomorphism in terms of lattice structures, we calculate the mean number of complements for all elements of a lattice. Because a lattice is defined as ($P(A)$, $\subseteq$), any elements of a lattice is a subset of A, $S\subseteq A$. A complement of S is defined as an element of $P(A)$, S^c, such that $S\cap S^c=\varnothing$ and $S\cup S^c=A$. For an element of a lattice, there can be no complement or plural complements. Therefore, the mean value of the number of complements for a lattice is expressed as a positive real value. For a distributive lattice, the mean value of complements is smaller than 1. Especially for a Boolean lattice, the value exactly coincides with a value 1.

Figure 4 shows time development of the mean value of the number of complements for a pair lattices representing type- and token-relations. It is clear to see that in type-computing Boolean lattice is robustly maintained although relations are perpetually changed and that in token-computing non-distributed lattice robustly appear. As a result dynamical duality between Boolean and non-distributive lattices are kept in time development.

The result shown in Figure 4 is very interesting in the context of brain model. In a distributive lattice, any parallel processing as lattice operations can be summed up with one operation by using a distributive law. On the other hand, in a non-distributive lattice, there are many cases plural parallel processing cannot be summed up. In assuming that summing up parallel processing as one processing based on a module is conscious operation, computing in distributive lattice can be regarded as conscious computing by manipulating symbols. It can be computing in linguistic area like Brocca and Wernicke area. By contrast, computing in non-distributive lattice can represent subconscious computing that cannot be explicitly appear in conscious mind. Dynamical duality between Boolean and non-distributive lattice can be regarded as genesis and maintenance of the duality of conscious cognition and subconscious perception, in an abstract sense.

4 Conclusion

We propose a toy model of phenomenal computing consisting an explicit computing system and its environments to execute computing. In general, a self-reference in the form of the mixture between computing and its environments is destined to be a contradiction. By contrast, self-referential form as a phenomenal computing can avoid a contradiction because of indefiniteness of environments. In our framework, the notion of indefinite environments is constructed as an infomorphism and self-reference is constructed as a mixture of relations and an infomorphism. Instead of resulting in a contradiction, the system perpetually changes the relationship between explicit computing and its environments.

Brain is a typical material that has to be regarded as a phenomenal computing. Especially focusing on difference between non-autistic and autistic children's reactions for Sally-Ann task, we propose an abstract model for a dynamical brain. Brain model consists of two sub-computing systems connecting by an infomorphism, and both sub-systems and an infomorphism are perpetually changed. A whole system in general results in a duality consisting of a computing system represented by Boolean lattice and one represented by non-distributive lattice. It is possible to see that the model mimics duality of conscious explicit cognition and subconscious implicit perception. Our model also explains difference between non-autistic and autistic children with respect to cognition of the relationship between parts and a whole.

References

Barwise, J. & Seligman, J. *Information Flow, The Logic of Distributed Systems*, Cambridge Univ. Press. 1997.

Chalmers, D.J. *The Conscious Mind: In Search of a fundamental Theory.* Oxford Univ.Press, 1998.

Dreyfus, H. *What Computers Still Can't Do: A Critique of Artificial Reason.* MIT Press, 1992.

Gunji, Y.-P., Takahashi, T. & Aono, M. Dynamical infomorpshim: Form of endo-perspective. *Chaos, Solitons & Fractals* (submitted).

Gunji, Y.-P., Miyoshi, H., Takahashi, T. & Kamiura, M. Dynamical duality of type- and token-computing as an abstract brain. *Chaos, Solitons & Fractals* (submitted).

Gunji, Y.-P. *What is My Own Consciousness?* Philosophy Press, Tokyo, 2003 (in Japanese).

Goldblatt, R. *Topoi, The Categorical Analysis of Logic.* North-Holland, Amsterdam, 1984=1991 (3rd).

Happe, F. Studying weak central coherence at low levels: children with autism do not succumb to visual illusion: a research note. *J.Child.Psychol.Psychiatry* 37:873-877 (1996).

Happe, F. Autism: cognitive deficit or cognitive style? *Trends Cogn.Sci.* 3:216-222 (1999).

Hermelin, B. & O'connor, N. Idiot savants calendrical calculators: rules and regularities. *Psychol. Med.* 16:885-893 (1986).

Kripke, S.A. *Wittgenstein, On Rules and Private Language*. Harvard univ. Press, 1982.

McCrathy, J. & Hayes, P.J. Some Philosophical Problems from the Standpoint of Artificial Intelligence, In: *Machine Intelligence 4*(B. Meltzer & D. Michie eds.), pp. 463-502, Edinburgh: Edinburgh Univ. Press, 1969.

Preston, J.and Bishop, M. *Views into the Chinese Room: New Essays on Searl and Artificial Intelligence.* Oxford Univ. Press, 2002.

Ramachandran, V.S. *Phantoms in the Brain.* William Morrow, New York, 1998.

Rimland, B. & Fein, D. In: *Special Talents of Autistic Savants in the Exceptional Brain* (L.K. Obler & Fein, D. eds.), The Guilford Press, New York, pp. 472-492 (1988).

Sacks, O. *The Man Who Mistook his Wife for a Hat.* Duckworth, London, 1985.

Sullivan, R.C. Rain man and Joseph. In: *High-functioning Individuals with Autism* (E. Schopler & Mesibov, G.B. eds.), pp. 243-251, Plenum Press, New York, 1992.

Selfe, L. *Nadia: A Case of Extraordinary Drawing Ability in Children.* Academic Press, London, 1977.

Snyder, A. and Mitchell, D.J. Is integer arithmetic fundamental to mental processing?: the minds's secret arithmetic. *Proc.Roy.Soc.Lond.* B266:587-592, 1999.

Tye, M. Ten Problems of Consiousness: *A Representational Theory of the Phenomenal Mind.* MIT Press, 1996.

Wimmer, H. & Perner, J. Beliefs about beliefs. *Cognition* 13:103-128 (1983).