인지과학, 제23권 제3호
Korean Journal of Cognitive Science
2012, Vol. 23, No. 3, 367～388.

Turing's Cognitive Science: A Metamathematical Essay for His Centennial

Woosik Hyun[†]

Hoseo University

The centennial of Alan Mathison Turing(23 June 1912 - 7 June 1954) is an appropriate occasion on which to assess his profound influence on the development of cognitive science. His contributions to and attitudes toward that field are discussed from the metamathematical perspective. This essay addresses (i)Turing's mathematical analysis of cognition, (ii)universal Turing machines, (iii)the limitations of universal Turing machines, (iv)oracle Turing machine beyond universal Turing machine, and (v)Turing test for cognitive science. Turing was a ground-breaker, eager to move on to new fields. He actually opened wider the scientific windows to the mind. The results show that first, by means of mathematical logic Turing discovered a new bridge between the mind and the physical world. Second, Turing gave a new formal analysis of operations of the mind. Third, Turing investigated oracle Turing machines and connectionist network machines as new models of minds beyond the limitations of his own universal machines. This paper explores why the cognitive scientist would be ever expecting a new Turing Test on the shoulder of Alan Turing.

Keywords : Turing, Turing Machine, Universal Turing Machine, Oracle Turing Machine, Gödel

[†] 교신저자: 현우식, 호서대학교 인문대학
E-mail: godel@hoseo.edu

Turing and Cognitive Science

The centennial of Alan Mathison Turing(23 June 1912 - 7 June 1954, Fig.1) is an appropriate occasion on which to assess his profound influence on the development of cognitive science. Turing's impact on the development of cognitive science was at once seminal and direct. His mathematical device now called *Turing machine* established bounds on what is and is not computable. He gave a remarkable definition of computing machine as human computer, which he employed in a way that looks very much like an information processing formal system as an intelligent system.

Since Turing’s mathematical model is regarded as the embodiment of mathematical thinking at the most fundamental level, the cognitive scientists’ claim on the Turing machine as a theoretical tool is as strong as the Artificial Intelligence group's one. That is, within a finite time, the Turing machine is capable of any computation that can be done by any super digital computer, no matter how powerful. Thus, for the theoretical study of the maximal problem-solving capacity, the Turing machine and its degree is a necessary condition. For example, the chess programs of C. Shanon and A. Turing; LISP of J. McCarthy; Logic Theorist and General Problem Solver of A. Newell, H. A. Simon and J. C. Shaw; PROLOG of F. Green, R. Kowalski and A. Colmerauer; SOAR of A. Newell, J. Laird, and P. Rosenbloom; Automated Mathematician Program of R. Davis and D. B. Lenat; Boyer-Moore theorem prover of R. S. Boyer and J. S. Moore, they are just some of the historical works worth noting in the models of cognitive science[1].

Figure 1. Alan Mathison Turing

In his 1936 paper[2], Turing claimed that it is possible to construct a machine to do the work of the human computer. According to Turing, to each state of mind of the human computer corresponds an *m-configuration* of the Turing machine.

> The behaviour of the computer [the human computer] at any moment determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to that the umber of states of mind which need be taken into account is finite. The reason for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape. ([2], p.136)

His idea is based on that the simple operations of the Turing machine must include:

(a) Changes of the symbol on one of the observed squares.

(b) Changes of one of the squares observed to another square within squares of one of the previously observed squares.

According to Turing, it may be that some of these changes necessarily involve a change of state of mind. The most general single operation must therefore be taken to be one of the following:

(A) a possible change (a) of symbol together with a possible change of state of mind.

(B) a possible change (b) of observed squares, together with a possible change of state of mind. ([2], p.137)

In his “Some Remarks on the Undecidability Results(1972),” Gödel stressed that Turing gives an argument that mental procedures cannot go beyond mechanical procedures[3]. Although Gödel accepted Turing’s analysis of the computability, he did not agree with Turing on this point. Gödel claimed that Turing’s argument is inconclusive:

> What Turing disregards completely is the fact that *mind, in its use, is not static, but constantly developing*, i.e., that we understand abstract terms more and more precisely as we go on using them, and that more and more abstract terms enter the sphere of our understanding. (Italics in original, [3], p.306)

In Gödel’s view, Turing disregarded time-factors of the mental capability. Actually, this claim stressed the possibility of designing machines that would be capable of learning. In contrast, Gödel himself focused on the developing process as a significant capability of cognition. Consequently, Gödel suggested two possibilities such as existence and convergence:

(1) There may exist systematic methods of actualizing this development;
(2) Turing’s number of *distinguishable states of mind* may *converge toward infinity*.

This process implies to form stronger and stronger axioms of infinity in set theory. According to Gödel’s account, such developing processes would decide a number-theoretic question. The existence of *finite non-mechanical procedures* is not excluded by Turing’s analysis.

> [T]he question of whether there exist finite *non-mechanical* procedures, not equivalent with any algorithm, has nothing whatsoever to do with the adequacy of the definition of “formal system” and of “mechanical procedure.” (Italics in

original, [4], p.370)

According to Gödel, the incompleteness results do not establish any bounds for the powers of human cognition, but rather than for the potentialities of pure formalism in mathematics.

Universal Turing Machines

A physical symbol system is an instance of a *universal Turing machine*. Thus, the development of the digital computers and automata theories originates with Turing's work in "On Computable Numbers, with an Application to the Entscheidungsproblem"[2]. Turing's work transformed the term *finite procedure* into *mechanical procedure*. Thus, a function is computable, or effectively calculable, if it can be calculated by a finite mechanical procedure, that is, by a Turing machine. Gödel claimed: "a formal system can simply be defined as any mechanical procedure for producing formulas, called provable formulas"[4]. In this sense, a function is Turing computable if it is definable by a Turing machine. According to this framework, a human mind as a Turing machine would yield m on input n if, when the mind is started on input n, it eventually halts, and at the moment when it halts, the tape represents m

The Turing machine is a finite automaton with unlimited tape as a memory device. It is mathematically equivalent to the class of the Herbrand-Gödel-Kleene equation system, that is, the class of general recursive functions[5,6]. Rather, Gödel endorsed the concept of Turing machines as a generally accepted property of effective calculability, not as general recursion defined by himself[7].

> The most satisfactory way, in my opinion, is that of reducing the concept of finite procedure to that of a machine with a finite number of parts, as has been done by the British mathematician Turing ([7], p.305).

> Turing's work gives an analysis of the concept of mechanical procedure (alias algorithm or computation procedure or finite combinatorial procedure). This concept is shown to be equivalent with that of a Turing machine ([4], p.369).

Historically, it is very remarkable that Gödel positively accepted Turing's thesis and analysis and, thereafter, always gave credit to Turing, not to Church([8], p.88):

> But I was completely convinced only by Turing's paper. (Gödel: Letter to Kreisel of May 1, 1968).

Turing's Thesis

If a function is informally computable, that is, definable by a finite mechanical procedure or algorithm, then it is computable by a Turing machine.

Turing's Thesis says that every algorithm can be programmed on a one-tape Turing machine. By Turing's thesis, we obtain the following:

1. If a function is definable by a finite mechanical procedure, then it is computable by a Turing's idealized human computer.
2. If a function is effectively calculable, then it is computable by a Turing's idealized human computer.
3. If a function is calculable by a Turing's idealized human computer, then it is *Turing computable*.

Turing devised an idealized human computing agent with the concepts of function produced by mechanical procedure. Turing's theorem states that any function calculable by an idealized human computer is Turing computable[2]. Furthermore, Turing's thesis asserts that if a function is informally computable, then it is computed by a Turing idealized human computer, meaning that every algorithm can be programmed on a one-tape Turing machine.

TURING MACHINE

Let S be a finite set of symbols including Blank 0 and Stroke 1, and let $q_1, q_2, .., q_n$ $(n \in N)$ be symbols of states not in S. Then a Turing machine on S is a finite set of quintuples

$$(q_i, s, t, \Phi, q_j),$$

where s and t are in S and Φ is one of the symbols R(move one right) or L (move one left), such that no two distinct quintuples have the same first two members. The symbol q_i represents the state i. Mathematically, a Turing machine is a function TM such that for some natural number n,

$$TM: \{0,1,2,...,n\} \times \{0,1\} \rightarrow \{0,1\} \times \{L, R\} \times \{0,1,2,...,n\}$$

where L stands for "move one left" and R "move one right".

The Turing machine is characterized by the following:

(1) a list of states called by Turing machine configurations: a specification of how many states there are.

(2) a finite alphabet of symbols, including blank and stroke.

(3) a finite set of lists of instructions. Each instruction has the form of the quintuple (i, s, t, Φ, j), where i and j are numbers no greater than the number of states, s and t are elements of the alphabet, and Φ is either *R*(move one right) or *L*(move one left). The instructions may be read: if in state i and scanning a cell containing s, then replace s with t ,move as Φ directs, and go into new state j.

Physically, a course of process can only consist of the following three steps(Fig.2):

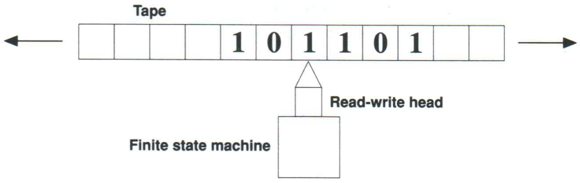


Figure 2. Physical Concept of Turing Machine

1. A symbol is written on the tape square being scanned, thereby erasing the previous symbol.
2. The Turing machine moves one square to the right or left.
3. The next state is specified.

Turing compared a human in the process of computing a natural number to a machine that is only capable of a finite number of conditions q_1, q_2, $\cdots$, q_r that is called "*m-configurations*." According to Turing, the behavior of the machine is determined by the *m-configuration* q_n and the scanned symbol s_r. This pair (q_n, s_r) is called the *configuration*. Thus, the configuration determines the possible behavior of the machine. Since there are only finitely many pairs, the behavior of the machine is specified by a finite list.

UNIVERSAL TURING MACHINE

For a recursive function F, there exists a universal Turing machine UTM such that

$$F_{TM_n}(x) = F_{UTM}(n,x)$$

for any Turing machine TM_n and for any natural numbers n and x. This is to say that there is a universal Turing machine that can simulate any Turing machine.

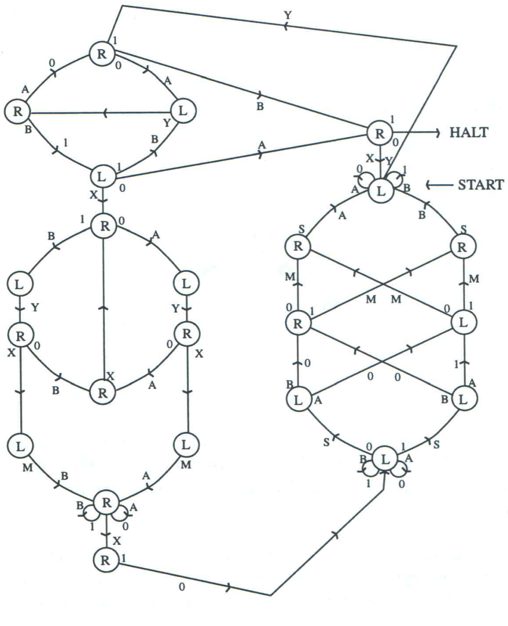


Figure 3. Universal Turing Machine (M. Minsky)

Universal Turing machine became fundamental for the theoretical conceptualization of modern general-purpose digital computers, as realized by Turing and von Neumann in the 1950s(Fig.3). Turing's analysis is a basis for cognitive science in that he began with modeling human cognition by carrying out a computation through a finite sequence of symbol manipulation.

The Limitation of Universal Turing Machines

Turing proved that Hilbert’s 10th problem, *Endscheidungsproblem*(decision problem), of discovering a method for establishing the truth or falsity of any statement in the first-order calculus, was impossible to solve by the Turing machine[2]. According to Gödel,

> In consequence of later advances, in particular of the fact that, due to A. M. Turing’s work, a precise and unquestionably adequate definition of the general concept of formal system can now be given, the existence of undecidable arithmetical propositions and the non-demonstrability of the consistency of a system in the same system can now be proved rigorously for *every* consistent formal system containing a certain amount of finitary number theory. ([4])

A sequence is said to be computable if it can be computed by a Turing machine. A number is computable if it differs by an integer from the number computed by a Turing machine. A function f is *Turing computable* if there exists a Turing machine TM that computes f. For an n-place function f, TM computes f if and only if, any $x_1,\ldots,x_n$ of the natural numbers, TM produces $f(x_1,\ldots,x_n)$ on input x. It is well known that Turing computable function f is a decidable $(n+1)$-ary relation and a recursively enumerable relation. A set A is recursively *enumerable* if A

is empty set or is the range of a general recursive function.

Turing, furthermore, demonstrated the limitations of Turing computability, proving that there are unsolvable problems, that is, the *Halting Problem*, in the Turing machine system[2]. There is a fundamental result in unsolvable problems: There exists a recursively enumerable set which is not recursive. The halting function for the Turing machine is a mechanical implementation of Gödel's undecidable sentences in the incompleteness theorems. Gödel himself credited that Turing's 1936 paper provides an adequate analysis of mechanical procedures and that, as a consequence of his work, a general formulation of the incompleteness theorems can be given[4]. Whether the halting problem is unsolvable by a Turing machine is a question concerning Turing machines themselves, prompting a metamathematical rather than a mathematical question.

LIMITATION OF UNIVERSAL TURING MACHINE

There is no Turing machine TM such that, for all e and n, if the Turing machine Gödel-numbered e produces something on input n then TM produces 0 on input (e,n); if the Turing machine Gödel-numbered e produces nothing on input n then TM produces 1 on input (e,n).([2]).

Turing claimed:

> For each of these "general process" problems can be expressed as a problem concerning a general process for determining whether a given integer n has a property $G(n)$ (e.g. $G(n)$ might mean "n is satisfactory" or "n is the Gödel representation of a provable formula"), and this is equivalent to computing a number whose n-th figure 1 if $G(n)$ is true and 0 if it is

false. ([2], p.134)

Both Gödel's and Turing's theorems show the limitations of the first-order calculus system or of recursive universal machines. From this point of view, Turing machines can compute only a proper subset of the functions. Thus, if the mind is a Turing machine and cognition is a Turing computable function, then the mind would not be able to compute such Halting functions, because they would not be in the class of cognition.

LIMITATION OF COGNITION AS TURING MACHINE

Let cognition COG be a Turing computable function on a natural number. Then there is non-computable cognition on the natural numbers.

We want to show that there exists a non-computable function in Turing computable functions over the natural numbers. Suppose that all Turing computable functions are computable in terms of Turing machine. Then there are a countable number of Turing computable functions $COG: \mathbb{N} \rightarrow \mathbb{N}$. Thus one can enumerate the functions $COG_1, COG_2, \ldots,$ where the way in which the computation depends upon n. And define a Turing computable function $COG: \mathbb{N} \rightarrow \mathbb{N}$ by

$$COG(n) = COG_n(n) + 1.$$

where $COG(n)$ is the action of some Turing machine on the natural number n as input. Since COG is a Turing computable function, it must appear somewhere in our list of computable functions.

Put $COG = COG_k$. But then

$$COG_k(k) \text{ if and only if } COG_k(k)+1,$$

which is a contradiction. Thus, COG is not computable.

Oracle Turing Machines beyond Universal Turing Machines

To overcome the computational limitation of the universal Turing machine, Turing proposed an extension of his machine model([9]). This idea gave rise to important issues such as *arithmetical hierarchy* and *relative recursiveness*([10]). When studying the relative recursiveness*(complexity)* of problems, we want to prove statements like: “Problem A is computable if any desired information about problem B is given for free.” Indeed, the notion can be considered as some kind of *reducibility*, and has been named *Turing reducibility* after its inventor.

ORACLE TURING MACHINE

An *oracle Turing machine* is simply a Turing machine with an extra “read only” tape, called the *oracle tape*, upon which is written the characteristic function of some

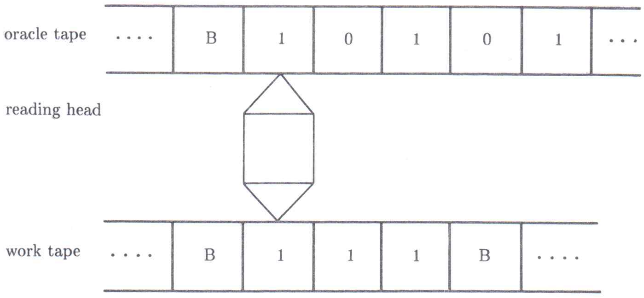


Figure 4. Oracle Turing Machine

set O called the *oracle*, and whose symbols cannot be printed over. The old tape is called the *work tape*. The reading head moves along both tapes simultaneously(Fig.4).

An Oracle Turing machine is a function OTM such that for some natural number n,

$$OTM: \{0,1,2,...,n\}\times\{0,1,2\}\times\{0,1\}\rightarrow\{0,1\}\times\{L, R\}\times\{0,1,2,...,n\}$$

where $\{0,1,2\}$ is the oracle tape alphabet, L stands for "move one left" and R for "move one right."

An oracle O for a function $f:\mathbb{N}\rightarrow\mathbb{N}$ is a device that, for a natural number $n\in\mathbb{N}$, responds the value $f(n)$. Suppose A and B are arbitrary sets, and for all $n\in\mathbb{N}$,

$$n\in A \leftrightarrow f(n)\in B.$$

Then, we have a decision procedure for membership in A if we have a decision procedure for membership in B. If there exists a decision procedure which computes $f(n)$ from n using an oracle O for B, for all $n\in\mathbb{N}$, then A is reducible to B through f, written $A \leq_f B$.

The oracle tape, therefore, is a query tape. An oracle O for B is an external agent that will supply the correct answer to questions of the form "$x\in B$" or not, for every $x\in\mathbb{N}$. We can replace f by a Turing machine if f is recursive. By this, we can consider the problem of relative computability.

Although the oracle has a new and powerful feature, it is the least constructive approach. It is remarkable to note that

(1) O is not necessarily identified with an algorithm,

(2) B may not be recursive,

(3) f may accept members of $\mathbb{N}^{\mathbb{N}}$ as arguments.

For a given formal system TM_1, one can add the statement $Cons(TM_1)$ of consistency of TM_1 as a new axiom to TM_1 in order to obtain TM_2. Similarly we can obtain $Cons(TM_2), Cons(TM_3), \ldots$ Turing's discoveries could imply that any true sentence is provable at some state in the transfinite iteration process. Thus, this process is clearly not complete within some finitary means.

The universal Turing machine is defined as the zero-order oracle Turing machine. Hence, the first-order oracle Turing machine is clearly more powerful than the previous Turing machines. However, it is also clear that the power comes from the addition of a function that was previously not computable. This leads to a recursive function that accepts members of the uncountably infinite set as inputs, which raises the problem of relative computations on recursive infinite functions.

Turing Test for Cognitive Science

Between the years 1948 and 1952 Turing had written and lectured on the question whether machines can be said to think. Thus, Turing proposed the question: "Can a computing machine behave as if it think?"[11]. His imitation game now called "Turing Test" for machine intelligence is to ask that Turing machine can normally answer questions from a human interrogator well enough to fool that interrogator into believing that the answers might be coming from a human being.

In terms of a game, the question is explained. The called imitation game is played by a man (A), a woman (B), and a human interrogator (C). The interrogator is in a room apart from the other two and tires to determine through conversation which of the other two is the man and which is the woman. Turing suggested that a teleprinter be used to communicate to avoid giving the interrogator clues through tones of voice. In this game the man may engage in deception in order to encourage the interrogator to misidentify him as woman. The man may lie about his appearance and preferences. Next Turing introduced his question.

> We now ask the question, 'What will happen when a machine takes the part of A in this game?' 'Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?' These questions replace our original, 'Can machines think?' ([11], p.434)

Still, the issue is left open as to exactly what passing the Turing test would mean. Yet it is clear that Turing test is neither a logically necessary nor logically sufficient condition for thinking. It is just a requirement for judging whether or not some systems do.

Many cognitive scientists have found Turing's assertion vital to continue Turing's arguments[16]. For example, D. Hofstadter and D. Dennett defended Turing's view[12]. Against Turing's viewpoint, R. Penrose remarkably revived the mathematical objection from Gödel's incompleteness theorems[13,14]. According to him, human mathematicians are not using a knowably sound algorithm in order to ascertain mathematical truth. Penrose maintains that human mind cannot be a computing machine.

According to Penrose, there are at least four viewpoints concerning the relationship between mathematical thinking and computation[14]:

A. All thinking is computation, in particular, feelings of conscious awareness are evoked merely by the carrying out of appropriate computations. (Alan Turing, Allen Newell, Herbert Simon, John McCarthy, Marvin Minsky, Douglas Hofstadter)

B. Awareness is a feature of the brain's physical action; and, whereas any physical action can be simulated computationally, computational simulation cannot by itself evoke awareness. (John Searle)

C. Appropriate physical action of the brain evokes awareness, but this physical action cannot even be properly simulated computationally. (Roger Penrose)

D. Awareness cannot be explained by physical, computational, or any other scientific terms. (Kurt Gödel)

Penrose himself is a proponent of viewpoint strong C that there should be something outside known physics, which means our current knowledge of physics is insufficient for a complete understanding of greater awareness. Penrose then analyzes that Turing's viewpoint is contained within A, that is, the so-called 'Strong AI' or 'computational functionalism', and Gödel's view in D, that is, the 'mystical category.'

Less convincing is Penrose's assertion that Gödel's viewpoint is mystical and Gödel's incompleteness theorems imply the negation of Turing's assertion. Although he uses Gödel's own statements as evidence, the reference is very limited in that it does not show that Gödel refuted viewpoint A. Contrary to Penrose's claim, Gödel's conclusive disjunction[15] can be viewed as being logically consistent not only with viewpoint D, but also with viewpoint A. The restriction of Turing's viewpoint looks close to Strong AI, but on the whole Turing's is more flexible. When Turing did speak of "the computer" in his 1936 paper, he always meant a human; he never meant machine. Note that even in 1946, Turing continued to use the term "computer" only about humans.

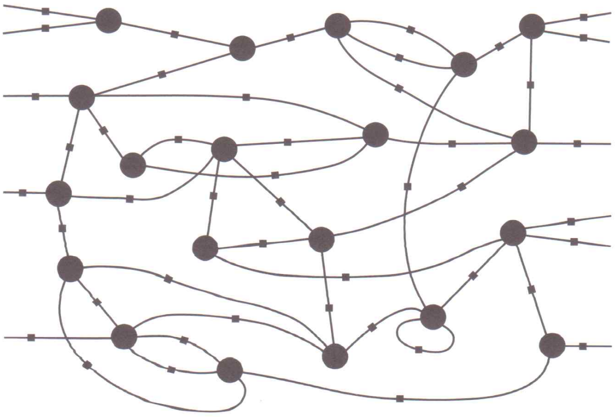
There exist two alternatives on the relation of the mind and the machine. In 1951, Gödel delivered the 25th Josiah Willard Gibbs Lecture, entitled *Some Basic Theorems on* *the Foundations of Mathematics and Their Implications*[15], at the American Mathematical Society. In this talk, Gödel addressed the significance of the incompleteness theorems for the controversies surrounding the nature of mathematics and the limitations of human cognition. He focused on the fundamental issue of the human mind and mechanical objects. Gödel asserted that the following disjunctive conclusion is inevitable with respect to the undecidable:

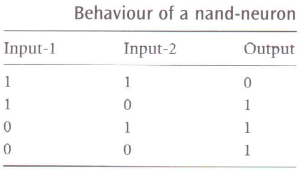
> *The human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite machine, or else there exist absolutely unsolvable diophantine problems of the type specified* (where the case that both terms of the disjunction are true is not excluded, so that there are, strictly speaking, three alternatives). (Italics in original, [15], p.310)

Gödel's disjunction is not exclusive or. Thus, we get both two alternatives as a true proposition. Gödel and I are in close agreement on this point for a big picture.

Concluding Remarks

Few realize that Turing already investigated neural network machine as early as 1948. In the summer of 1948 Turing completed a report describing the outcomes of his research into 'how much a machine can do for the higher' part of the brain. The unpublished report "Intelligent Machinery(1948)" as Turing's prophecy introduced many concepts including *expert system*, *genetic algorithm*, *neural network*(Fig.5), and *connectionism*[17]. Unfortunately, the relation of computability to brain, including the physical basis of mind, is left by Turing's sudden death. In the early 1950's Turing ventured the field of artificial life, using nonlinear differential equations to express the





Behaviour of a nand-neuron

Input-1	Input-2	Output
1	1	0
1	0	1
0	1	1
0	0	1

Figure 5. Turing's Neural Network

chemistry of growth.

Turing's orientation was predominantly metamathematical rather than mathematical. A metamathematical point of view, due to Turing, as to what objects and methods of argument in cognitive science should be counted as mathematically reliable. The main requirements of Turing's cognitive science imply:

(1) the object of arguments are provable objects.

(2) the operations that can be applied are defined and can in principle be computable.

(3) the assertion that there exists an object x with the property $cognitve(x)$ means that one can either produce a concrete example of such an object or show a way of configurating one.

As a whole, Turing was a ground-breaker, eager to move on to new fields. He actually opened wider the scientific windows to the mind. First, by means of mathematical logic Turing discovered a new bridge between the mind and the physical world. Second, Turing gave a new formal analysis of operations of the mind. Third, Turing investigated oracle Turing machines and connectionist network machines as new models beyond the limitations of his own universal machines. The cognitive scientist would be ever expecting a new Turing Test on the shoulder of Alan Turing.

References

[1] Herken, R., ed. (1988). *The Universal Turing Machine*, Oxford University Press, Oxford.

[2] Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, 42, 230-265.

[3] Gödel, K. (1972). Some remarks on the undecidability results, K. Gödel (1990), *Collected Works II: Publications 1938-1974*, S. Ferferman et al.(eds), Oxford University Press, Oxford.

[4] Gödel, K. (1964). Postscriptum to Gödel 1934, K. Gödel (1986), *Collected Works I: Publications 1929-1936*, S Ferferman et al.(eds), Oxford University Press, Oxford.

[5] Gödel, K. (1934). On undecidable propositions of formal mathematical systems, K. Gödel (1986), *Collected Works I: Publications 1929-1936*, S. Ferferman et al.(eds), Oxford University Press, Oxford.

[6] Kleene, S. (1952). *Introduction to Metamathematics*, D. Van Nostrand, Princeton.

[7] Gödel, K. (1951). Some basic theorems on the foundations of mathematics and their implications, K. Gödel (1995), *Collected Works III: Unpublished Essays and Lectures*, S. Ferferman et al.(eds), Oxford University Press, Oxford.

[8] Sieg, W. (1994). Mechanical Procedures and Mathematical Experience*, Mathematics and mind*, A. George (ed.), Oxford University Press, Oxford.

[9] Turing, A. (1939). Systems of logic based on ordinals, *Proceedings of the London Mathematical Society*, ser. 2, 45, 161-228.

[10] Soare, R. (1987). *Recursively Enumerable Sets and Degrees*, Springer-Verlag, New York.

[11] Turing, A. (1950). Computing Machinery and Intelligence, *Mind* 59, 433-460. in [17].

[12] Hofstadter D. and D. Dennett (1981). *The Mind's I*, Basic, New York.

[13] Penrose, R. (1989). *The Emperor's New Mind*, Oxford University Press, Oxford.

[14] Penrose, R. (1994). *Shadows of the mind*, Oxford University Press, Oxford.

[15] Gödel, K. (1951). Some basic theorems on the foundations of mathematics and their implications, K. Gödel (1995), *Collected Works III: Unpublished Essays and Lectures*, S. Ferferman et al.(eds), Oxford University Press, Oxford.

[16] Copeland, B. (1998). Turing's O-Machines, Searle, Penrose, and the Brain, *Analysis*, Vol.58, No.2, 128-138.

[17] Copeland, B., ed. (2004). *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus the Secrets of Enigma*, Oxford University Press, Oxford.

1 차원고접수 : 2012. 6. 7
2 차원고접수 : 2012. 7. 13
최종게재승인 : 2012. 8. 10

(요 약)

튜링의 인지과학:
튜링 탄생 백주년을 기념하는 메타수학 에세이

현 우 식

호서대학교

이 연구는 튜링의 탄생 백주년을 맞이하여 인지과학을 위한 그의 심대한 공헌을 고찰하기 위한 작업이다. 이 논문에서는 특히 튜링에게 가장 중요한 학문적 영향을 주었던 괴델의 시각을 통하여 튜링의 공헌과 입장이 논의된다. 이를 위하여 메타수학적 접근이 시도되며, (1) 튜링의 인지에 대한 수학적 분석, (2) 보편튜링기계, (3) 보편튜링기계의 한계, (4) 보편튜링기계의 한계를 넘는 모델로서의 오라클튜링기계, (5) 인지과학을 위한 튜링테스트가 논의된다. 이 연구에 의하면, 튜링의 공헌은 다음과 같이 정리될 수 있다. 첫째 튜링은 수리논리를 사용하여 마음과 물리적 세계의 새로운 가교를 발견했다. 둘째, 튜링은 마음의 작동에 대하여 새로운 형식적 분석을 제공했다. 셋째, 튜링은 자신의 튜링기계의 한계를 넘어서는 마음의 새로운 모델로서 오라틀 튜링기계와 연결주의적 신경망기계를 제시했다. 우리 인지과학자들은 튜링의 어깨 위에 서서 늘 새로운 튜링테스트를 기다리고 있게 될 것이다.

주제어 : 튜링, 튜링기계, 보편튜링기계, 오라클튜링기계, 괴델