Vol.2010-AL-131 No.12
2010/9/22

ゲーデル及びチューリング 再考

田 中 榮 一

ゲーデルの第一不完全性定理は，算術の体系に証明も反証もできない有限長論理式がある，と主張している．本文ではゲーデルが定理の証明に用いた論理式は本質的に無限長であることを明らかにした．ゆえに第一不完全性定理の証明は正しくない．しかし，本質的に無限長である論理式は証明も反証もできないから，本質的に無限長である論理式を持つ体系は不完全である．算術の体系にも本質的に無限長である論理式があるから，算術の体系は不完全である．チューリング機械の計算を表す論理式も本質的に無限長であるから，停止問題が決定不能であるのは当然である．

Reflections on Gödel and Turing

Eiichi Tanaka [†1]

Gödel's first incompleteness theorem on Peano arithmetic states that there exists a finite length formula that can be neither proved nor refuted. This paper clarifies that the formula that Gödel used in his proof is an essentially infinite length formula. This means that the proof of the first incompleteness theorem is incorrect. Nevertheless, the incompleteness theorem holds in any formal system with essentially infinite length formulas. Since the arithmetic includes essentially infinite length formulas, it is incomplete. The formula to express the halting problem of a Turing machine is also an essentially infinite length formula. Therefore it is natural that the halting problem is undecidable.

1. Introduction

Gödel's incompleteness theorems published in 1931 [1)] have been considered as one of the epoch making discoveries in the history of mathematics. The theorems have given immeasurable influence not only on the view of mathematics but also on human thoughts. Gödel studied Peano arithmetic and found that if the arithmetic is ω consistent, there exists a formula that we can neither prove nor refute. This is called the 1st incompleteness theorem. Inspired by Gödel's paper, Turing [2)] studied the halting problem of a computing machine that he proposed, namely a Turing machine. He proved that the halting problem of a Turing machine is undecidable.

In this paper we shall show that the provability predicate is an essentially infinite length formula. That is, Gödel's proof of the first incompleteness thorem is incorrect. Any formal systems with essentially infinite length formulas are incomplete. Furthermore, the predicate to express the halting problem of a Turing machine is also an essentially infinite length formula. Therefore the undecidability of the halting problem is intuitively understood.

†1 Kobe University

2. Preliminaries

The predicate logic in this paper follows Shoenfield [3)], but the classification of symbols is slightly modified. The symbols of the predicate logic for the arithmetic are defined as follows. (a1) individual constants $(a, b, c, \cdots)$, (a2) variables $(x, y, z, \cdots)$, (a3) function symbols $(+, *)$, (a4) a predicate symbol $(=)$, (a5) logical symbols 1 ($\neg, \vee$), (a6) a logical symbol 2 ($\exists$), (a7) subsidiary symbols ((,), comma). $\exists$ is called an existential quantifier. In this paper let the basic symbols of the predicate logic for the arithmetic be the symbols of (a1) $\sim$ (a5) and (a7).

Let A and B be sets of collections of objects. A mapping from the set of n-tuples in A to B is called an n-ary function from A to B. A subset of the set of n-tuples in A is called an n-ary predicate in A. An occurrence of x in predicate A is bound in A, if it occurs in a part of A of the form $\exists x A$, otherwise it is free in A.

Definition 1.
[1] An individual constant is a term.
[2] A variable is a term.

[3] If $t_1, \cdots, t_n$ are terms and f^n is an n-ary function, $f^n(t_1, \cdots, t_n)$ is a term.

Definition 2.
[1] If $t_1, \cdots, t_n$ are terms and P is an n-ary predicate, $P(t_1, \cdots, t_n)$ is a formula.
[2] If A and B are formulas, $A \vee B$ and $\neg A$ are formulas.
[3] Let $\{A_0, A_1, A_2, \cdots\}$ be an infinite set of formulas. Define $R = A_0 \vee A_1 \vee A_2 \vee \cdots$. R is a formula.
[4] If $C(x)$ is a formula and x is a free variable, $\exists x C(x)$ is a formula.

Formulas $(A \rightarrow B), (A \wedge B), (A \leftrightarrow B)$ and $\forall x A(x)$ are the abbreviations of $(\neg A \vee B), \neg(A \rightarrow \neg B), ((A \rightarrow B) \wedge (A \leftarrow B))$ and $\neg\exists\neg A(x)$, respectively. Symbol $\forall$ is a universal quantifier. A formula without free variables is called a closed formula or a sentence. We sometimes define function symbols and predicate symbols that are not in the basic symbols. In this paper we assume that defined function symbols and defined predicate symbols are rewritten using the basic symbols.

Sometimes $\exists x C(x)$ represents finite numbers of Cs such that $C(x)(x = 0, 1, \cdots, n)$. We call such an $\exists x C(x)$ a finite existential formula. If $\exists x C(x)$ represents infinite numbers of Cs such as $C(x)(x = 0, 1, \cdots)$, we call $\exists x C(x)$ an infinite existential formula.

Definition 3.
The length of a function is defined as the number of the basic symbols in the function. If a function consists of infinitely many basic symbols, it is called an infinite length function. If a function is not an infinite length function, it is a finite length function. The length of a formula and that of a predicate are similarly defined.

Formula R in Definition 2 is an infinite length formula. Let Q be a finite length formula without infinite existential formulas. Assume that $\exists x C(x)$ is an infinite existential formula and there is an axiom or a theorem such that $\exists x C(x) \rightarrow Q$. $\exists x C(x)$ can be converted to a finite length formula.

Definition 4.
If an infinite length formula can not be transformed into a finite one with any efforts, the formula is called an essentially infinite length formula.

We introduce Bochvar's ternary logic. The truth values of each formula are T, F and M, which indicate true, false and meaningless, respectively. $H_\vee(A, B)$ and $H_\neg(A)$ are ternary truth functions which have the truth values of $A \vee B$ and $\neg A$, respectively. $H_\vee(A, B)$ and $H_\neg(A)$ are defined as follows.

$$H_\vee(T, T) = H_\vee(T, F) = H_\vee(F, T) = T$$
$$H_\vee(F, F) = F$$
$$H_\vee(T, M) = H_\vee(M, T) = H_\vee(M, M) = M$$
$$H_\neg(T) = F$$
$$H_\neg(F) = T$$
$$H_\neg(M) = M$$

In this paper value M is assigned unconditionally to an essentially infinite length formula.

Five inference rules are introduced.
(1) Expansion Rule. Infer $B \vee A$ from A
(2) Contraction Rule. Infer A from $A \vee A$
(3) Association Rule. Infer $(A \vee B) \vee C$ from $A \vee (B \vee C)$
(4) Cut Rule. Infer $B \vee C$ from $A \vee B$ and $\neg A \vee C$
(5) $\exists$ -Introduction Rule. If x is not free in B, infer $\exists x A \rightarrow B$ from $A \rightarrow B$

Note that in this paper these rules are not applied to formulas with value M. For instance, if formula A has value M, $B \vee C$ can not be infered from $A \vee B$ and $\neg A \vee C$ by Cut Rule.

An expression is a sequence of symbols. If an expression consists of infinite symbols, it is called an infinite length expression. If not, it is a finite length expression. The concept of Gödel number is interesting and useful. However, there is a dicrepancy between the Gödel number of a predicate and its length. As we shall see later, the predicate

Vol.2010-AL-131 No.12
2010/9/22

$\exists z T(x, y, z)$ for the halting problem of a Turing machine is an example. That is, the Gödel number of the predicate is finite, but its length is essentially infinite. To get rid of this deficit we must realize that the Gödel number of a predicate is finite if and only if its length is finite. This aim is achieved by excluding a quantifier. This is why the basic symbols do not include an existential quantifier.

Many types of Gödel numbering have been proposed. We do not specify which one we use. Gödel numbering satisfies the following characteristics.

(*1) Different finite sequences of finite length expressions have different Gödel numbers.

(*2) There is an algorithm to decide whether a given number is the Gödel number of a finite sequence of finite length expressions or not. If it is so, we can find the finite sequence of finite length expressions.

Consider the following two functions.

$$G_1(x) = S(x) \tag{1}$$

$$G_2(x) = g_0(x) * S(0) + g_1(x) * S(1) + g_2(x) * S(2) + \cdots \tag{2}$$

where $g_k(x)$ is a function such that if $x = k$, $g_k(x) = 1$, and otherwise, $g_k(x) = 0$. $G_1(x)$ and $G_2(x)$ are the functions with the same values, but their expressions are different. The Gödel number of $G_1(x)$ is finite and that of $G_2(x)$ is infinite. Gödel numbering is only based on an expression. $G_2(x)$ is not an essentially infinite length function, because that $G_2(x)$ can be rewritten to a finite length function $G_1(x)$.

The number of infinite length formulas is infinite, and the Gödel number of an infinite length formula is also infinite. Therefore, if a formula is an infinite length formula, we can not reconstruct the original formula from its Gödel number.

(*3) A finite sequence of finite length expressions satisfies (*1) and (*2).

(*4) If a finite sequence of expressions contains at least one infinite length expression, it does not satisfy (*1) and (*2).

Remark 1.

Gödel numbering is effective for a finite sequence of finite length expressions. If a finite sequence of expressions contains at least one infinite length expression, Gödel numbering is not effective.

3. Infinite Length Formulas

We would like to remark on essentially infinite length formulas. Any essentially infinite length formula φ can be neither proved nor refuted, because that a proof must be done in finite steps. Then we have the followings.

$$S \nvdash \varphi, \qquad S \nvdash \neg\varphi \tag{3}$$

(3) corresponds to the 1st incompleteness theorem.

Let π be a finite length formula. $\pi \vee \neg\pi$ is tautology. That is,

$$S \vdash \pi \vee \neg\pi \tag{4}$$

As we have seen in (3), the values of φ and $\neg\varphi$ are not defined in the sence of a binary logic. Therefore, nobody can tell that $\varphi \vee \neg\varphi$ is tautology. Hence we introduce Bochvar's ternary logic. Let $V(A)$ be the value of formula A. We have the following.

$$V(\varphi \vee \neg\varphi) = M \qquad V(\varphi \wedge \neg\varphi) = M \tag{5}$$

where $\varphi \vee \neg\varphi$ means the consistency of S, and $\varphi \wedge \neg\varphi$ does the inconsistency of S. We can prove neither the consistency nor the inconsistency of S. (5) corresponds to the 2nd incompleteness theorem.

Remark 2.

Any formal theory is incomplete and its consistency and inconsistency can not be proved, if the system has at least one essentially infinite length formula.

Peano arithmetic includes essentially infinite length formulas. Then, from Remark 2 we have the following.

Remark 3.

The arithmetic is incomplete and its consistency can not be proved.

Vol.2010-AL-131 No.12
2010/9/22

4. Diagonal Sequences

(1) Diagonal Sequences

The set of finite length formulas is a countably infinite set. Enumerate all finite length formulas with one free variable u.

$$A_0(u), A_1(u), A_2(u), \cdots \tag{6}$$

Consider formulas $I(u)$ and $J(u)$ such as

$$I(k) = \neg A_k(k)(k = 0, 1, 2, \cdots). \tag{7}$$

$$J(k) = A_k(k)(k = 0, 1, 2, \cdots). \tag{8}$$

(7) and (8) are called the antidiagonal sequence of (6) and the diagonal sequence of (6), respectively. If $I(u)$ is in (6), it is a finite length formula. It is easy to prove by the diagonal method that $I(u)$ is not in (6). Since $I(u)$ is not in the set of all finite length functions (6), it is an essentially infinite length formula. Furthermore, we have

$$J(u) = \neg I(u). \tag{9}$$

Since $I(u)$ is an essentially infinite length formula, so is $\neg I(u)$. From (9), $J(u)$ is not in (6). That is, $J(u)$ is an essentially infinite length formula.

Change the order of formulas in (6). Let it be as follows.

$$A'_0(u), A'_1(u), A'_2(u), \cdots \tag{10}$$

Let $I'(u)$ and $J'(u)$ be the antidiagonal sequence of (10) and the diagonal sequence of (10), respectively. $I'(u)$ and $J'(u)$ are also essentially infinite length formulas. Note that there are infinitely many different sequences defined by all finite length formulas with one free variable. For each sequence, there are an antidiagonal sequence and a diagonal sequence. Both of them are essentially infinite length sequences. If an antidiagonal sequence and a diagonal sequence are defined based on all finite length formulas with one free variable, we need not pay attension to the order of formulas.

Lemma 1.

Any antidiagonal sequence and any diagonal sequence defined by all finite length formulas with one free variable are essentially infinite length formulas.

(2) Unprovable predicates

Consider predicate $\mathbf{Sb}(\mathbf{x}, \lceil z \rceil, Z(y))$, where predicate $\mathbf{x}$ has a free variable z, $\lceil z \rceil$ is the Gödel number of z, $Z(y)$ is the numeral denoting the number y. $\mathbf{Sb}(\mathbf{x}, \lceil z \rceil, Z(y))$ indicates the predicate obtained by substituting $Z(y)$ for z in $\mathbf{x}$. For simplicity we use notation $D_{\lceil \mathbf{x} \rceil}(y)$ instead of $\mathbf{Sb}(\mathbf{x}, \lceil z \rceil, Z(y))$.

Assume that each D is a finite length formula and has one free variable y. Enumerate all Ds in the following way.

$$D_0(y), D_1(y), D_2(y), \cdots \tag{11}$$

$\mathbf{Sb}(\mathbf{x}, m, Z(x))$, that is $D_x(x)$, has values $D_k(k)(k = 0, 1, 2, \cdots)$. Define formula $H(x)$ as follows.

$$H(k) = D_k(k) \quad (k = 0, 1, 2, \cdots) \tag{12}$$

Note that predicate $\mathbf{x}$ in $\mathbf{Sb}(\mathbf{x}, m, Z(x))$ can be any formula $A_k(u)$ of (6), where $k = 0, 1, 2, \cdots$. That is, $A_k(u)$ is expressed as $\mathbf{Sb}(\mathbf{A_k}, \lceil z \rceil, Z(u))$. The $(k+1)$-th formula of $J(u)$ is written as

$$J(k) = \mathbf{Sb}(\mathbf{A_k}, \lceil z \rceil, Z(k)) \quad (k = 0, 1, 2, \cdots) \tag{13}$$

Therefore, $I(u)$ and $J(u)$ can be expressed using framework $\mathbf{Sb}(\mathbf{x}, \lceil z \rceil, Z(x))$. Since $H(u)$ includes all formulas of (6), $H(u)$ satisfies the condition of Lemma 1, that is, "a diagonal sequence defined by all finite length formulas with one free variable".

In the similar way of the proof of Lemma 1 we can prove that $H(u)$ is an essentially infinite length formula. Therefore it is very natural that $H(u)$, namely $\mathbf{Sb}(\mathbf{u}, m, Z(u))$, can be neither proved nor refuted. Changing notations from $\mathbf{u}$ and u to $\mathbf{x}$ and x, respectively, we have the following.

Lemma 2.

Predicate $\mathbf{Sb}(\mathbf{x}, m, Z(x))$ is an essentially infinite length predicate and can be neither proved nor refuted.

Gödel defined predicate $\mathbf{x}B_\kappa\mathbf{y}$, where κ is any set of formulas, $\mathbf{y}$ is a formula that we are going to prove, and $\mathbf{x}$ is a proof of $\mathbf{y}$. B_κ is a primitive recursive predicate. If $\mathbf{a}$ is

$\mathbf{Sb}(\mathbf{y}, m, Z(y))$, $\mathbf{x}B_\kappa\mathbf{a}$ is an infinite length predicate.

Lemma 3.
Predicate $\mathbf{x}B_\kappa\mathbf{Sb}(\mathbf{y}, m, Z(y))$ is an essentially infinite length predicate and can be neither proved nor refuted.

Define the provability predicate.

$$Bew(\mathbf{y}) \equiv (\exists\mathbf{x})\mathbf{x}B_\kappa\mathbf{y} \tag{14}$$

The right-hand side (14) means that there exists a proof $\mathbf{x}$ for $\mathbf{y}$, and the predicate is called the provability predicate. $Bew(\mathbf{y})$ indicates that $\mathbf{y}$ is provable. In order to understand the problem in the proof of the incompleteness theorems, we shall quote Gödel's words [1]. Some of the notations and definitions are not defined here, but we can refer to his paper for them. Several symbols are changed, for instances, from ligthface to boldface and (x) to $\forall x$ etc.

We refer to Heijenoort's English translation [4] of Gödel's paper [1]. Gödel stated as follows.

[1] (Heijenoort, pp.606-607)

"Theorem V. For every recursive relation $R(x_1, \cdots, x_n)$ there exists an n-place RELATION SIGN $\mathbf{r}$ (with the FREE VARIABLES $u_1, \cdots, u_n$) such that for all n-tuples of numbers $(x_1, \cdots, x_n)$ we have

$$R(x_1, \cdots, x_n) \rightarrow Bew[Sb(\mathbf{r}, u_1, \cdots, u_n, Z(x_1), \cdots, Z(x_n))] \tag{15}$$

$$\overline{R}(x_1, \cdots, x_n) \rightarrow Bew[Neg(Sb(\mathbf{r}, u_1, \cdots, u_n, Z(x_1), \cdots, Z(x_n)))] \text{ "} \tag{16}$$

[2] (Heijenoort, p.608)

" We obviously have

$$\forall\mathbf{x}[Bew_\kappa(\mathbf{x}) \sim \mathbf{x} \in Flg(\kappa)] \tag{17}$$

and

$$\forall\mathbf{x}[Bew(\mathbf{x}) \rightarrow Bew_\kappa(\mathbf{x})] \tag{18}$$

We now define the relation

$$Q(\mathbf{x}, \mathbf{y}) \equiv \overline{\mathbf{x}\mathbf{B}_\kappa[\mathbf{Sb}(\mathbf{y}, 19, Z(y))]} \tag{19}$$

Since $\mathbf{x}\mathbf{B}_\kappa\mathbf{y}$ are recursive, so is $Q(\mathbf{x}, \mathbf{y})$. Therefore, by Theorem V and (18) there is a RELATION SIGN $\mathbf{q}$ such that

$$\overline{\mathbf{x}\mathbf{B}_\kappa[\mathbf{Sb}(\mathbf{y}, 19, Z(y))]} \rightarrow Bew_\kappa[\mathbf{Sb}(\mathbf{q}, (17, 19), (Z(x), Z(y))] \tag{20}$$

and

$$\mathbf{x}B_\kappa[\mathbf{Sb}(\mathbf{y}, 19, Z(y))] \rightarrow Bew_\kappa[Neg(\mathbf{Sb}(\mathbf{q}, (17, 19), (Z(x), Z(y)))]. \text{ "} \tag{21}$$

$\overline{R}$ and $NegP$ indicate $\neg R$ and $\neg P$, respectively. The numbers 17 and 19 are the Gödel numbers of certain free variables.

There is a serious mistake in the above description. Gödel claims that $Q(\mathbf{x}, \mathbf{y})$ is recursive. From Lemma 3, $\mathbf{x}B_\kappa[\mathbf{Sb}(\mathbf{y}, 19, Z(y))]$ is an essentially infinite length predicate. Therefore $Q(\mathbf{x}, \mathbf{y})$ is not recursive. Hence Theorem V can not be applied to $Q(\mathbf{x}, \mathbf{y})$. That is, there does not exist a recursive relation sign $\mathbf{q}$. Thus, Gödel's argument is wrong.

The first incompleteness theorem claims that there exists at least one finite length formula that can be neither proved nor refuted. However Lemma 2 and Lemma 3 mean that the proof of the first incompleteness thorem is incorrect.

(3) The halting problem [2),5)]

The next topic is the halting problem of a Turing machine. The halting problem is described as follows. Can we decide whether or not there is a computation of a given Turing machine with given any data ? Let x, y and z be the Gödel number of a given Turing machine Tm, a given data and the Gödel number of a computation, respectively. Predicate $\exists zT(x, y, z)$ means that there is a computation z for a given Turing machine x and a given data y, where T is a primitive recursive predicate. The halting problem is the decision problem of predicate $\exists zT(x, y, z)$. It is well known that since predicate $\exists zT(x, x, z)$ is recursively undecidable, the halting problem is recursively undecidable.

Let $T_k(u)$ be the Gödel number of the Turing machine that computes the representing function $f_k(u)$ of $A_k(u)$ in (6). Then, $\exists zT(T_k(u), k, z)$ indicates the computation

Vol.2010-AL-131 No.12
2010/9/22

of $f_k(k)$. Therefore, the framework of $T(a,b,c)$ can treat all the representing functions of finite length formulas. Let $F_x(y)$ be $\exists zT(x,y,z)$. $F_x(x)$ satisfies the condition of Lemma 1.

From the viewpoint of this paper we must evaluate the length of $\exists zT(x,y,z)$. We can prove that $F_x(x)$ is an essentially infinite length formula by the similar way of the proof of Lemma 1. Therefore, it is very natural that $\exists zT(x,x,z)$ is undecidable. So $\exists zT(x,y,z)$ is undecidable.

Lemma 4.
Predicate $\exists zT(x,y,z)$ is an essentially infinite length predicate and undecidable.

5. Concluding Remarks

We can summarize the conclutions just obtained as follows:
(1) A diagonal sequence and an antidiagonal sequence defined by all finite length formulas with one free variable are essentially infinite length formulas.
(2) The provability predicate is an essentially infinite length predicate. Therefore Gödel's proof of the first incompleteness theorem is incorrect. Nevertheless, the incompleteness theorem holds in any formal system with infinite length formulas. Since the arithmetic includes essentially infinite length formulas, the arithmetic is incomplete.
(3) The predicate for the halting problem of a Turing machine is an essentially infinite length predicate. Therefore, it is very natural that the halting problem is undecidable.

Acknowledgment
We would like to express our thanks to sencere metamathematicians for their comments. Both constructive and critical comments were very stimulative.

References

1) Gödel, K.; "Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme,", *Monatshefte für Mathematik und Physik*, 38, pp.173-198, (1931)
2) Turing, T.; "On Computable Numbers, with an Application to the Entscheidungsproblem,", *Proceedings of the London Mathematical Society*, ser.2, 42, pp.230-265, (1936)
3) Shoenfield, J. R.; *Mathematical Logic*, Addison-Wesley, Massachusets, (1967)
4) Heijenoort, J.; *From Frege to Gödel*, Harvard University Press, Massachusetts, (1967)
5) Davis, M.; *Computability and Unsolvability*, McGraw-Hill, NY, (1958)