Common sense in Yamashita Lab.

Gödel's incompleteness theorems

Yukihiko Yamashita
Tokyo Institute of Technology

Contents

- What is ‘set’
- Gödel's first incompleteness theorem
- Gödel's second incompleteness theorem

1 Set

Queistion: **What is set?**

Queistion: $X = \{S \mid S \textbf{ is a set}\}$ is a set?

1.1 Russell's paradox

Let $W = \{S \mid S \notin S\}$.

Queistion: $W \in W$**?**

Assume that $W \in W$, then $W \notin W$.
Assume that $W \notin W$, then $W \in W$.

Axioms of set is necessary

1.2 ZermeloFraenkel set theory with axiom of choice

- Axiom of extensionality: $\forall z(z \in x \leftrightarrow z \in y) \leftrightarrow x = y$
- Axiom of pairing: $\exists z \forall u(u \in z \leftrightarrow u = x \text{ or } u = y)$
- Axiom of union: $\exists y \forall z(z \in y \leftrightarrow \exists u(u \in x \text{ and } z \in u))$
- Axiom of power set $\exists y \forall z(z \in y \leftrightarrow z \subset x)$
- Axiom of empty set: $\exists x \forall y(y \notin x)$
- Axiom of infinity: $\exists x[\phi \in x \text{ and } \forall y(y \in x \rightarrow y \cap \{y\} \in x)]$
- Axiom schema of replacement:
  $\forall x \forall y \forall z(\varphi(x, y) \text{ and } \varphi(x, z) \rightarrow y = z)$
  $\rightarrow \exists u \forall y[y \in u \leftrightarrow \exists x(x \in u \text{ and } \varphi(x, y))]$
- Axiom of regularity: $x \neq \phi \rightarrow \exists y(y \in x \text{ and } y \cap x \neq \phi)$
- (Axiom of choice):
  $\forall x \in u(x \neq \phi) \text{ and } \forall x, y \in u(x \neq y \rightarrow x \cap y = \phi)$
  $\rightarrow \exists v \forall x \in u \exists! t(t \in x \text{ and } t \in v)$

We can discuss almost all mathematical issues.
Queistion: Is ZFC is consistent?

1.3 Peano axioms

Because ZFC is too difficult, we show axioms of natural numbers called Peano axioms.

- Any natural number x has its successor x'.
- There exist a natural number 0 that is not a successor.
- If $x' = y'$, we have $x = y$.
- For any logical expression $\varphi(x)$, we have the following relation

$$\varphi(0), \forall x(\varphi(x) \to \varphi(x')) \leftrightarrow \forall x \varphi(x)$$

  (Mathematical induction).

1.4 Recursive functions

In order to define calculation on integers, the primitive recursive function is defined.

- Function which provide a constant c:

$$f(x_1, \ldots, x_k) = c.$$

- Function which select a input variable x_i $(1 \leq i \leq k)$:

$$f(x_1, \ldots, x_k) = x_i.$$

- Function which provides the successor:

$$f(x) = x'.$$

- Compound function: Assume that $f(x_1, \ldots, x_k)$, $g_1(x_{11}, \ldots, x_{1n_1})$,$\ldots$, and, $g_k(x_{k1}, \ldots, x_{kn_k})$ are primitive recursive functions,

$$f(g_1(x_{11}, \ldots, x_{1n_1}), \ldots, g_k(x_{k1}, \ldots, x_{kn_k})).$$

- Function defined recursively: Assume that $g(x_2, \ldots, x_k)$, $h(x, y_1, \ldots, y_k)$

are primitive recursive functions.

$$f(0, x_2, \ldots, x_k) = g(x_2, \ldots, x_k)$$
$$f(x', x_2, \ldots, x_k) = h(f(x, x_2, \ldots, x_k), x, x_2, \ldots, x_k)$$

Example: For $g(x) = x$, $h(x) = x'$

$$\text{plus}(0, y) = g(y)(= y)$$
$$\text{plus}(x', y) = h(\text{plus}(x, y))(= \text{plus}(x, y)')$$

Question: What is $\text{plus}(x, y)$**.**

Let's prove $\text{plus}(x, y) = \text{plus}(y, x)$ **by using Peano axioms.**

Hint: Prove $\text{plus}(0, y') = \text{plus}(0, y)'$, $\text{plus}(x, y') = \text{plus}(x, y)'$, and $\text{plus}(x, y') = \text{plus}(y', x)$.

Recursive function $f(x_1, \ldots, x_k)$: $t = f(x_1, \ldots, x_k)$: is defined as the minimum t such that $g(x_1, \ldots, x_k, t) = 0$ with a primitive recursive function $g(x_1, \ldots, x_k, t)$,

Calculation is defined as a function realized by a recursive function.

2 Incompleteness theorems

2.1 First and second Incompleteness theorems

Assume that T is a recursive and formal theory **including natural numbers.**

(i) If T is consistent, there exists a sentence G in T such that **we cannot prove either G or $\neg G$.**

(ii) If T is consistent, **the consistence of T cannot proven.**

- $\neg G$: the negation of G.
- G is called **Gödel's sentence.**
- G can be realized.

2.2 Gödel's sentence

- We can map a natural number to every sentence. Thus there is one to one mapping between a sentence and a natural number.
- For example, if we express a sentence by character codes, it is a huge binary number.
- For a sentence A, the corresponding natural number is described by $\ulcorner A \urcorner$ and called the Gödel's number of A.
- Let $\mathrm{Bew}(\ulcorner A \urcorner)$ be a logical expression such that A can be proven.
- Since we can give a natural number to a proof, "There is a proof" is equivalent to "There is a natural number with a condition".
- Since **a primitive recursive function can check whether a number is a proof or not**, we can know whether a proof exists or not by a recursive function.
- Proofs can be handled in the natural number theory.
- Let $\mathrm{Sub}(n, m)$ be the Gödel's number of a logical expression such that a natural number n is substituted into a free variable of a logical expression of Gödel's number m, **which has only one free variable.** This is a func-

tion from two integers to a integer and realized by a primitive recursive function.

- A logical expression $R(n)$ is defined by $\neg\text{Bew}(\text{Sub}(n,n))$.
- The Gödel's sentence G is defined by $R(\ulcorner R(n) \urcorner)$.
- In $\ulcorner R(n) \urcorner$, a Gödel's number as a free variable is assigned to n.
- We have the following equivalency:

$$G \leftrightarrow \neg\text{Bew}(\text{Sub}(\ulcorner R(n) \urcorner, \ulcorner R(n) \urcorner)) \tag{1}$$

- This $\text{Sub}(\ulcorner R(n) \urcorner, \ulcorner R(n) \urcorner))$ is given by the Gödel's number of the logical expression when we substitute the Gödel's number of $R(n)$ into n in $R(n)$. Therefore it is the Gödel's number of $R(\ulcorner R(n) \urcorner)$ or G. (Note that $\ulcorner R(n) \urcorner$ is a natural number.)
- Then, we have the following equivalency:

$$G \leftrightarrow \neg\text{Bew}(\ulcorner G \urcorner) \tag{2}$$

This means **if G is true, we cannot prove G, and if G is not true, we can prove.**

2.3 Consistence

Definition Theory T is consistent:

There is no case such that **both A and $\neg A$ are proven** for a logical expressions A.

- If a theory T is not consistent, we can prove $A \wedge \neg A$ for a logical expression A.
- For any logical expression B, from a logic theory we have

$$\neg B \rightarrow A \vee \neg A (= \text{true}).$$

- By considering its contraposition, we have

$$A \wedge \neg A \rightarrow B. \tag{3}$$

- Since $A \wedge \neg A$ can be proven, **any logical expression B is proven.**
- Therefore, **not consistent theory can not use at all.**
- $\text{Con}(T)$ denotes that a theory T is consistent.

3 Outline of proof of Gödel's first incompleteness theorem

- If G can be proven, G becomes true and G cannot be proven. This contradicts to that G can be proven.

- If $\neg G$ can be proven, we can prove $\mathrm{Bew}(\ulcorner G \urcorner)$. Then, $\mathrm{Bew}(\ulcorner G \urcorner)$ is true and G can be proven. Because G and $\neg G$ can be proven, This contradicts the consistence of the theory.

4 Outline of proof of Gödel's second incompleteness theorem

Let's describe formally what we proved for the first theorem:

$$\mathrm{Con}(T) \rightarrow \neg\mathrm{Bew}(\ulcorner G \urcorner)$$
$$\mathrm{Con}(T) \rightarrow \neg\mathrm{Bew}(\ulcorner \neg G \urcorner)$$

If we can prove $\mathrm{Con}(T)$ formally, we can prove $\neg\mathrm{Bew}(\ulcorner G \urcorner)$ and can prove G. This contradicts to the first theorem.

5 Conclusion

- Diagonal method is used.
- The concept ‘calculation’ that is very important in computer was born to discuss theories and proofs strictly.
- In order to show there is no proof, we have to define procedure or algorithm.
- For the purpose,the recursive function and the Turing machine are defined, and computers are invented.
- Anyway I feel uneasy since we cannot prove the consistence of present theories of mathematics.

Proof of $\text{plus}(x, y) = \text{plus}(y, x)$.

1. We will prove $\text{plus}(x, 0) = \text{plus}(0, x)$ for any x by mathematical induction with respect to x. When $x = 0$, we have

$$\text{plus}(0, 0) = \text{plus}(0, 0).$$

Assume that $\text{plus}(x, 0) = \text{plus}(0, x)$, we have

$$\text{plus}(x', 0) = (\text{plus}(x, 0))' = (\text{plus}(0, x))' = x' = \text{plus}(0, x').$$

2. We will prove $\text{plus}(x, y') = \text{plus}(x, y)'$ for any x and y by mathematical induction with respect to x. When $x = 0$, we have for any y

$$\text{plus}(0, y') = y' = \text{plus}(0, y)'$$

Assume that $\text{plus}(x, y') = \text{plus}(x, y)'$ for any y, we have

$$\text{plus}(x', y') = \text{plus}(x, y')' = (\text{plus}(x, y)')' = \text{plus}(x', y)'$$

3. Now will will prove $\mathrm{plus}(x, y) = \mathrm{plus}(y, x)$ for any x and y by mathematical induction with respect to y. 1. yields

$$\mathrm{plus}(x, 0) = \mathrm{plus}(0, x).$$

Assume that $\mathrm{plus}(x, y) = \mathrm{plus}(y, x)$ for any x, we have from 2.

$$\mathrm{plus}(x, y') = (\mathrm{plus}(x, y))' = (\mathrm{plus}(y, x))' = \mathrm{plus}(y', x).$$