Incompleteness and Computabilits

BO-YU Cheng (鄭伯俞), Department of Electrical Engineering , National Chiao Tung University, Taiwan (tiny.nemo.fishgmail.com)
Advisor: Prof. Ming-Hsuan Kang (康明軒)

Introduction

Gödel's Incompleteness Theorems are famous theorems in Mathematical Logic. In this article , I present a proof of *Gödel's First Incompleteness Theorems* based on concepts from *Computability Theory*. The result suggests that *incompleteness* happens not only in *First Order Peano Arithmatic* but also in any Formal Theory that has the capability to reason about programs.

Definitions

- A **Formal Theory** T consists of :

[1] A finite set of symbol Σ_T

[2] The set of finite sequences of Σ_T ,(denoted by Σ_T^*), is called *strings in T*

[3] Elements in a decidable subset of Σ_T^* are called the **formulas** in T

[4] A *formula* **x** may be considered to be a **proof** of another *formula* **y** (" *is_a_proof_of* " is a binary relation on *formulas*)

[5] A formula f is **provable** if there exists a proof of f

- Given a *Formal Theory* T, **programs** in T are the computable functions with *input* and *output* in Σ_T^*.We can also encode a program P as some *string* $[P]_T$ in T , such that another program U can simulate the behaviour of P by given $[P]_T$ as input.
- For any Formal Theory T , a *program* **V** is said to be a **Proof Verifier** for **T** if $V(\phi, p)$ decides whether p is a proof of ϕ in T.
- We say a Formal Theory T can **reason about programs** if :

[1] T has a negation sign "¬"

[2] There exists some computable mapping H such that if a program P halts on input x with output y , then

  - $\mathbf{H}([\mathbf{P}]_\mathbf{T}, \mathbf{x}, \mathbf{y})$ is a provable formula .
  - $\neg\mathbf{H}([\mathbf{P}]_\mathbf{T}, \mathbf{x}, \mathbf{z})$ is a provable formula for all $z \neq y$

- . Given a Formal Theory T that reasons about programs:
  - We say that T is **consistent** if there is no formula ϕ in T such that both ϕ and $\neg\phi$ are provable.
  - We say that T is **complete** if for any formula ϕ in T , either ϕ is provable or $\neg\phi$ is provable.

Theorem

Theorem 1. *Let T be a formal theory with a proof verifier V ,which can reason about programs , then T cannot be both complete and consistent.*

Proof. Consider the following program M in T , whose input is the encoding $[X]$ of another program X (if the input is not an encoding , we don't care what M does):

Algorithm 1 Program M $([X]_T)$
Require: Proof Verifier V of T
  for all *formula* p in T **do**
    if $V(H([x],[x],0)\ ,p)$ **then** ▷ p proves that $X([X])$ returns 0
      return 1
    else if $V(\neg H([X],[X],0)\ ,p)$ **then** ▷ p proves that $X[X]$ doesn't return 0
      return 0
    end if
  end for

Now , Assume T is consistent. Consider the result of M([M]). There are 3 cases :

[1] $H([M],[M],0)$ is provable in T, and $\neg H([M],[M],0)$ is not provable in T.

[2] $\neg H([M],[M],0)$ is provable in T, and $H([M],[M],0)$ is not provable in T.

[3] Both $H([M],[M],0)$ and $\neg H([M],[M],0)$ are not provable in T

By the definition of M , in case 1 , we have $M([M])$ returns 1. Since T reasons about programs , $H([M],[M],1)$ is provable in T , which contradicts the assumption that T is consistent. Similarly , in case 2 , we have $M([M])$ returns 0. Since T reasons about programs , $H([M],[M],0)$ is provable in T , which contradicts the assumption that T is consistent. Hence , only case 3 can happen , which by definition means T is incomplete. □

[t]1.0

Remark

[1] **Relation to Gödel's Original Statement** : The proof presented here is actually more general than Gödel's original statement , in the sense that *PA* is a special case of of our *formal theory*.In fact , Gödel proved a lemma in his original proof (often referred to as $\beta - lemma$) , which essentially proofs that *PA* can *reason about programs*.

[2] **Interpretation About the Result** This proof shows that Incompleteness not only occurs in *PA* , but also Formal Theorys that have the ability to reason about program.

Reference

[1] A computability-base-proof of 1'st incompleteness Theorem. *https://www.scottaaronson.com/blog/?p=710*

[2] Computability based Proof of 2'nd incompleteness theorem. *http://www.ams.org/notices/201011/rtx101101454p.pdf*