Incompleteness Theorem

CS5616 - Computational Complexity

Neeraj Krishna N
112101033

April 27, 2024

Introduction

Theorem (Incompleteness Theorem)

1. *In any formal system, there are true statements that cannot be proven*

Introduction

Theorem (Incompleteness Theorem)

1. *In any formal system, there are true statements that cannot be proven*
2. *Any formal system cannot prove its own consistency, unless it is inconsistent*

Definitions

Definition (Formal System)

Any system with a set of axioms and a set of inference rules is called a formal system

Example : Peano Arithmetic

Definitions

Definition (Formal System)

Any system with a set of axioms and a set of inference rules is called a formal system

Definition (Consistency of a formal system)

A formal system is inconsistent if it can prove a statement and its negation. A formal system which is not inconsistent is said to be consistent

Definitions (Contd.)

Definition (Soundness)

All provable statements must be true

Definitions (Contd.)

Definition (Soundness)

All provable statements must be true

Definition (Completeness)

All true statements are provable

Assumptions

We assume soundness, since it's not interesting to work with proof systems that are not sound i.e, we don't want proof systems to prove a false statement.

Assumptions

We assume soundness, since it's not interesting to work with proof systems that are not sound i.e, we don't want proof systems to prove a false statement.
In addition, we also assume the following properties[1]

1. Mathematical statements that can be precisely described in English should be expressible in the system

Assumptions

We assume soundness, since it's not interesting to work with proof systems that are not sound i.e, we don't want proof systems to prove a false statement.

In addition, we also assume the following properties[1]

1. Mathematical statements that can be precisely described in English should be expressible in the system
2. Proofs should be detailed and convincing, and they should be easy to check step by step

Proof of 1st Incompleteness Theorem

Let us call the set of provable statements as $\mathcal{P}$ and set of all true statements as $\mathcal{T}$

Proof of 1st Incompleteness Theorem

Let us call the set of provable statements as $\mathcal{P}$ and set of all true statements as $\mathcal{T}$

Claim

$\mathcal{P}$ is recursively enumerable

Proof of 1st Incompleteness Theorem

Let us call the set of provable statements as $\mathcal{P}$ and set of all true statements as $\mathcal{T}$

Claim

$\mathcal{P}$ *is recursively enumerable*

Claim

$\mathcal{T}$ *is not recursively enumerable*

Another interesting Proof

Let us look at another interesting proof using Kolmogorov Complexity [2]
We say that a pair $(\langle M \rangle, y)$ represents x if TM M on input y outputs x

Another interesting Proof

Definition (Kolmogorov Complexity)

Kolmogorov Complexity $K(x)$ of a bit string x is the smallest k such that there exists a representation $(\langle M \rangle, y)$ of x such that $|(\langle M \rangle, y)| \leq k$

In other words, we can think $K(x)$ to be the best possible compression of the string x. Note that this is well defined.

Proof using Kolmogorov Complexity

Let C be a compression algorithm and D be its corresponding decompression algorithm such that $\forall x \in \{0,1\}^*, D(C(x)) = x$
Then the representation of x is $(\langle D \rangle, C(x))$

Claim

$\forall x, |C(x)| \geq K(x) - O(1)$

Proof using Kolmogorov Complexity

There are strings whose Kolmogorov complexity is no smaller than the length of the string itself

Proof using Kolmogorov Complexity (Contd.)

Claim (Existence of Incompressible Strings)

For every n, there is a string $x \in \{0,1\}^n$ such that $K(x) \geq n$

Proof using Kolmogorov Complexity (Contd.)

Claim (Existence of Incompressible Strings)

For every n, there is a string $x \in \{0,1\}^n$ such that $K(x) \geq n$

Claim

For every n and every c, the probability that a random n-bit string x has Kolmogorov complexity $\geq n - c$ is more than $1 - \frac{1}{2^c}$

Proof using Kolmogorov Complexity (Contd.)

Definition

Let $\mathcal{R}$ be a set defined as follows
$\mathcal{R} = \{x \mid K(x) \geq |x|\}$

Note that this set is well defined by claim 3

Proof using Kolmogorov Complexity (Contd.)

Definition

Let $\mathcal{R}$ be a set defined as follows
$\mathcal{R} = \{x \mid K(x) \geq |x|\}$

Note that this set is well defined by claim 3

Theorem

$\mathcal{R}$ is not decidable

Generating Unprovable True Statements

Let us consider any formalization of mathematics where the following assumptions hold

1. For every binary string x and integer k, we can construct a statement $S_{x,k}$ equivalent to "$K(x) \geq k$"
2. If there is a valid proof P of a statement S, then S is true (Soundness)
3. It is decidable whether a given P is a valid proof of statement S

Generating Unprovable True Statements

Lemma

For every formalization of mathematics as described above, there is a threshold value t, such that all statements of the form $S_{x,k}$ with $k > t$ are unprovable

Generating Unprovable True Statements

Lemma

For every formalization of mathematics as described above, there is a threshold value t, such that all statements of the form $S_{x,k}$ with $k > t$ are unprovable

Consider the following algorithm, which takes k as the input and outputs a string $\in \{0,1\}^*$

1. $m \leftarrow 1$
2. while(True):
    1. For all strings x of length atmost m
       For all strings P of length atmost m
       If P is a valid proof of $S_{x,k}$, output x and halt
    2. $m \leftarrow m + 1$

Generating Unprovable True Statements

The previous algorithm gives us a way to generate statements that, with high probability are unprovable. How ??

Generating Unprovable True Statements

The previous algorithm gives us a way to generate statements that, with high probability are unprovable. How ??

1. Suppose we fix a formalization of mathematics (any formalization), then the algorithm above is well defined and can be written down. Let c be the number of bits, required to write it down.

Generating Unprovable True Statements

The previous algorithm gives us a way to generate statements that, with high probability are unprovable. How ??

1. Suppose we fix a formalization of mathematics (any formalization), then the algorithm above is well defined and can be written down. Let c be the number of bits, required to write it down.
2. Now we can find a k, such that $c < k - \log k$

Generating Unprovable True Statements

The previous algorithm gives us a way to generate statements that, with high probability are unprovable. How ??

1. Suppose we fix a formalization of mathematics (any formalization), then the algorithm above is well defined and can be written down. Let c be the number of bits, required to write it down.
2. Now we can find a k, such that $c < k - \log k$
3. Then we can be sure that, no statement of the form '$K(x) \geq k$' is provable.

Generating Unprovable True Statements

The previous algorithm gives us a way to generate statements that, with high probability are unprovable. How ??

1. Suppose we fix a formalization of mathematics (any formalization), then the algorithm above is well defined and can be written down. Let c be the number of bits, required to write it down.
2. Now we can find a k, such that $c < k - \log k$
3. Then we can be sure that, no statement of the form '$K(x) \geq k$' is provable.
4. Let us take a random string of length $k + 1$, then by claim 4, string is such that $K(x) \geq k$ with high probability, but such a statement is unprovable in our formalization.

A puzzle

Surprise Examination Paradox

the teacher announces in class: “next week you are going to have an exam, but you will not be able to know on which day of the week the exam is held until that day.” The exam cannot be held on Friday, because otherwise, the night before the students will know that the exam is going to be held the next day. Hence, in the same way, the exam cannot be held on Thursday. In the same way, the exam cannot be held on any of the days of the week.

Relation with 2nd Incompleteness Theorem

It can be argued [3], that the flaw in the derivation of the paradox is that it contains a hidden assumption that one can prove the consistency of the mathematical theory in which the derivation is done which is impossible by the second incompleteness theorem.

References

- L. Trevisan, "Notes on unprovable statements." https://theory.stanford.edu/~trevisan/cs172-07/notelogic.pdf.
- L. Trevisan, "Notes on kolmogorov complexity." https://lucatrevisan.github.io/teaching/cs172-15/notek.pdf.
- S. Kritchman and R. Raz, "The surprise examination paradox and the second incompleteness theorem." https://www.ams.org/notices/201011/rtx101101454p.pdf.

Thank You