Lob's Theorem: a brief introduction

Mark Xu

May 29, 2019

Abstract

In mathematical logic, Godel's Completeness Theorem says that statements are true if and only if they are provable. In a startling plot twist, Godel also proved a set of Incompleteness Theorems, the first of which states that for any sufficiently powerful mathematical system, there exist statements that cannot be proved or disproved. Given that nearly all mathematical systems that we care about are thus essentially incomplete, one might hope that they are sound in the sense that statements that are provable are true. While this is indeed the case, it is not the case that this soundness can be captured from within the system. Indeed, a theorem of Lob states that, if, for any statement σ, one can prove that "if σ is provable, then σ", then σ must be true. In this talk, we discuss the semantic notion of truth in comparison to the syntactic notion of provability. We then provide a sketch of Godel's First Incompleteness theorem and its implications, concluding with a proof of Lob's Theorem and a discussion on the various ways a system can fail to know itself.

1 Background

Mathematics is often thought of the study of Truth free from subjectivity. This is not entirely true (although it is not entirely false either). It would be more correct to say that mathematics is the study of Truth relative to a certain set of axioms. It would be even more correct to say that mathematics is the study of Truth relative to a certain set of axioms under a given deductive system. It would be the most correct to say that mathematics is the study of Truth relative to a certain set of axioms under a given deductive system *within a given structure*.

For everything that follows, we will be working with the axioms of Peano Arithmetic and it's corresponding language, but the results hold with more generality.

1.1 Models, Structures and Truth

Informally, a structure of a given language is a set of objects along with a set of interpretations of all the symbols in the language in relation to the given objects. For example, if your language is $\{S\}$, a possible interpretation is that your set is $\mathbb{N}$ and $S : x \mapsto x + 1$. A different structure is that your set is $\mathbb{R}$ and $S : x \mapsto x/2$. However, most of the time we want our structures to have, well, structure. The way we do this is to get a set of axioms A and constrain all structures so that our axioms are 'true' inside the structures. What do I mean by 'true'? Well, informally all mathematical statements are built from pieces connected together with logical connectives - we assign truth values to all of the base pieces (via our structure and/or deductive system) and then our statement is true if all of the right pieces are true. "Snow is white" is true if and only if snow is white. If a given set of axioms A is true inside a structure S, then we say that S models A and we write $S \models A$.

If a given sentence σ is such that for all structures S, if $S \models A$, then $S \models \{\sigma\}$, then we write that $A \models \sigma$.

1.2 Provability

Slightly separately from truth, we have this notion of provability. Informally, something is provable from a set of axioms if you can write down a proof for it. What's a proof? It's just a sequence of mathematical statements where each statement is either an axiom or follows from one of your rules of inference. What are these rules of inference? It turns out that the precise deductive system you use isn't really that important, it just has to include stuff like modus ponens.

Notice that this notion of provability only depends on your set of axioms and doesn't reference any particular model. If a sentence σ is provable from a set of axioms A, we write $A \vdash \sigma$.

1.3 Godel's Completeness Theorem

Theorem 1.1 (Godel's Completeness)**.** *For any set of axioms A, $A \vdash \sigma \iff A \models \sigma$.*

The proof is kind of tedious so I'm not going to explain it, but Godel's Completeness Theorem basically says that that if something is true is all possible models of a given set of axioms, then it must logically follow from those axioms and thus must have a proof. If this weren't the case, then there would be statement that were true in every world where a set of axioms were true, but we wouldn't be able to show that that was the case, which would make math kind of hard.

2 Godel's First Incompleteness Theorem

The standard English phrasing of Godel's First Incompleteness Theorem goes "There are statements in Peano Arithmetic that are true, but not provable". If you're using 'true' to mean 'true' in all models, then this phrasing is wrong. The obvious problem is that by Godel's Completeness Theorem, since PA is a set of axioms, all statements that are true in PA must be provable in PA. Some people use the word 'true' to mean 'true in the natural numbers', and in that sense, Godels' theorem does say that there are statements that are true but not provable. But you wouldn't expect them to be provable, because Godel's Completeness Theorem.

Theorem 2.1 (Godel's First Incompleteness)**.** *For any set of sufficiently powerful axioms A, there exists a mathematical statement such that A does not model σ and A does not model $\neg\sigma$. We call such a σ undecidable.*

What do I mean by 'sufficiently powerful'? Basically, you need to be able to encode Turing Machines inside A. For now, you can think of it as being able to prove the axioms of PA, but PA is actually much stronger than you need for this sort of thing.

2.0.1 Coding

Let's say that you're writing computer program and you need to use lists, but for some reason you really don't like lists. Since you really like natural numbers (who doesn't?), you come up with a clever way to use natural numbers to represent lists using the prime factorization theorem.

Given a list of numbers $(n_1, ..., n_k)$, we encode them in the product $\prod_i p_i^{n_i+1}$, where p_i is the ith prime number. Of course, encoding numbers means that you can encode any list drawn from any countable set: in particular, you can encode the language of arithmetic. Since a arithmetic statement is just a list of symbols with certain properties, with a lot of work, you can define what it means to be mathematical statement in arithmetic. Since a proof is just a list of statements that satisfy other properties, you can also define what it means to be a proof in arithmetic. Thus, with a whole lot of work, you can define a formula PROOF(n, s) := "n is the code of a proof of the sentence whose code is s". Using this, we define a formula PRVB(s) := $\exists n$: PROOF(n, s). For notation, given a formula ϕ, let $[\phi]$ be it's code.

Proof of Godel's First Incompleteness Theorem

Proof. Let's try the obvious thing and figure out why its wrong. Let $\varphi(x) := \neg$PRVB(x). Let c be the code of φ and consider $\varphi(c) \iff \neg$PRVB(c). Ideally, this statement $\varphi(c)$ would say something like "I am not provable", but it doesn't quite do that. Notice that $\varphi(c)$ says that φ is not provable, which doesn't really make any sense since φ needs a variable.

How do we get around this? Well the easy answer is something called the Diagonal Lemma, which states that for all formulas F, one can construct a sentence σ such that $\sigma \iff F([\sigma])$. If this Lemma is true, then we can let $\phi := \neg$PRVB(), and we're done. We now prove the Diagonal Lemma.

Let F be the property we want to diagonalize. First, we need to define a function $subs(n) = [\alpha(n)]$, where $[\alpha] = n$. In fact, what we actually need is a formula $issub(n, m) \iff subs(n) = m$. Such a formula exists because we have assumed our axiom system to be sufficiently powerful.

Given such a formula, define

$$\beta(z) = \forall y : (issubs(z, y) \implies F(y))$$

. Thus we have that our axiom system proves

$$\beta([\phi]) \iff \forall y : (y = [\phi([\phi])] \implies F(y)) \iff F([\phi([\phi])])$$

. Now we just plug in β for ϕ and we get

$$\beta([\beta]) \iff F([\beta([\beta])])$$

□

2.1 Non-Standard Models

So it seems like we can make some statement σ that says "I am not provable". OF course, since σ is actually not provable, it seems like this statement is somehow "true". The problem is that while being provable is not dependent on the model, trying to express provability in a formula causes such a dependence to arise. The trouble is when you know that PRVB(σ) is true and you want to extract the proof out of it. Ideally, you would get back some natural number that encoded the proof and then just convert it to symbols, but the problem is that this assumes that you're only quantifying over the natural numbers. If you have a strange model of arithmetic, then you might get back some transcendental number, and when you try to get the proof out, you might end up with an infinite sequence of:

$$...\neg\neg\neg\neg\neg\neg\sigma, \neg\neg\neg\neg\sigma, \neg\neg\sigma, \sigma$$

This makes it possible to have a model of arithmetic where Godel's statement is false. What if we tried to get rid of the 'non-standard' part of arithmetic? This turns out to be impossible

Theorem 2.2. *There is no set of axioms A that can only be modeled by $\mathbb{N}$*

Proof. Suppose such an A exists. Then Godel's sentence σ would be true in all possible models of A. By Godel's completeness theorem, $A \models \sigma$, a contradiction. □

This particular proof of Godel's First Completeness theorem also gives us Godel's Second Incompleteness Theorem for free.

Theorem 2.3 (Godel's Second Inconsistency)**.** *No consistent sufficiently strong recursive axiomatic system can prove that it is consistent.*

The first thing we have to figure out is if it's possible to express consistency in A. Well, if A was inconsistent, then there would be a proof of everything. Thus, consistency of A is biconditional with there being a statement that A cannot prove. We can thus define a consistency predicate $CON(A) := \neg\text{PRVB}([0 \neq 0])$. This allows us to talk about A's consistency within A.

Proof. Suppose that we could prove our system consistent. If we could prove our Godel statement, that would be a contradiction. Since we can prove that our system is consistent, we can thus prove that that we cannot prove our Godel statement. However, by definition of our Godel statement, this constitutes a proof of Godel's statement, a contradiction. Thus our system is not consistent. □

Godel's Second Incompleteness Theorem say that no system can prove its own consistency *in general*. However, we can hope that we might be able to prove consistency for specific formulas. While it is the case that PRVB() is globally unreliable, it might be the case that there are some statements σ such that A can prove $\text{PRVB}(\sigma) \implies \sigma$, i.e. when A is right whenever A thinks that it's proved σ. It turns out that this is not the case.

3 Lob's Theorem

Theorem 3.1 (Lob)**.** $(PA \models (PRVB(\sigma) \implies \sigma)) \implies (PA \models \sigma))$

Before we prove this theorem, we go over some properties of PA that we'll need.

3.1 Preliminaries

i $(PA \models \sigma) \implies (PA \models \text{PRVB}(\sigma))$

ii $PA \models (\text{PRVB}(\sigma) \implies \text{PRVB}(\text{PRVB}(\sigma)))$

iii $PA \models (\text{PRVB}(\sigma \implies \phi) \implies (\text{PRVB}(\sigma) \implies \text{PRVB}(\phi)))$

The key to proving Lob's theorem is a construction called Lob's Sentence. Basically, given any statement σ, our Lob's Sentence $L \iff (\text{PRVB}(L) \implies \sigma)$. This can be done by applying the diagonal lemma

Proof.

By definition of L	$PA \models (\text{PRVB}(L) \iff \text{PRVB}(\text{PRVB}(L) \implies \sigma))$	(1)
Hypothesis	$PA \models (\text{PRVB}(\sigma) \implies \sigma)$	(2)
By iii	$PA \models (\text{PRVB}(\text{PRVB}(L) \implies \sigma) \implies (\text{PRVB}(\text{PRVB}(L)) \implies \text{PRVB}(\sigma))$	(3)
By (1) and (3)	$PA \models (\text{PRVB}(L) \implies (\text{PRVB}(\text{PRVB}(L)) \implies \text{PRVB}(\sigma))$	(4)
By ii	$PA \models (\text{PRVB}(L) \implies \text{PRVB}(\text{PRVB}(L)))$	(5)
By (4), (5) & MP	$PA \models (\text{PRVB}(L) \implies \text{PRVB}(\sigma))$	(6)
By (2), (6), & MP	$PA \models (\text{PRVB}(L) \implies \sigma)$	(7)
By i & (7)	$PA \models \text{PRVB}(\text{PRVB}(L) \implies \sigma)$	(8)
By (1) & (8)	$PA \models \text{PRVB}(L)$	(9)
By (7) & (9)	$PA \models \sigma$	(10)

□