Lecture 10:
Incompleteness Theorem

Valentine Kabanets

October 25, 2016

1 Gödel's Incompleteness Theorems

1.1 Historical context

According to Kant, there were two main schools of philosophical thought in Europe in the 17th century: rationalists (Descartes, Spinoza, Leibniz) and empiricists (Hobbes, Locke). The rationalists believed that, in principle, all knowledge can be gained by reason alone. In contrast, the empiricists argued that all knowledge has to come from experience through senses. The former took mathematics, while the latter preferred physics as their model for knowledge.

Intuitively, the rationalists were saying that one could fix some finite collection of obvious truths (axioms), and some obviously valid rules of inference so that any true fact can be derived (reasoned out) from the axioms using the given inference rules.

The rationalists were trying to apply these principles to all possible domains. But what about mathematics itself? Can we axiomatize mathematics so that every true statement can be mechanically derived (reasoned out) from a fixed set of axioms using some fixed set of rules? What about some subset of mathematics such as arithmetic (a part of mathematics that studies various properties of natural numbers)?

This is the question seriously considered by a number of leading mathematicians and logicians in the late 19th and early 20th century, working on the *foundations of mathematics*. For example, the great German mathematician David Hilbert strongly believed that such axiomatization of mathematics exists. However, Kurt Gödel in 1930's argued that it is *not possible* to axiomatize arithmetic. That was a shocking discovery! It showed that even a formal domain such as arithmetic is too complex for every true fact to be gained by reason (deduction) alone.

We provide more details of Gödel's famous result next.

1.2 Setting up the stage

Informally, Gödel's Incompleteness Theorem says:

> *There are true arithmetic sentences whose truth cannot be proved.*

To make sense of this statement (and make it more precise), we need to explain the notions of *arithmetic sentence*, *truth*, and *provability*.

Arithmetic sentence: An *arithmetic formula* is a formula involving variables, arithmetic operations $+$, $*$, equality $=$, inequality $<$, constants $0, 1, 2, \ldots$, as well as the usual logical connectives (AND, OR, NOT), and quantifiers $\forall$ and $\exists$. The variables in the formula that are not quantified over are called *free variables*. A *sentence* is a formula without free variables. (For example, $2+2 = 4$ and $\forall x \exists y(x + 1 = y)$ are (true) arithmetic sentences.)

Truth: Every arithmetic sentence is either true or false, when interpreted over the set $\mathbb{N}$ of natural numbers, using the standard interpretation of $+$,$*$, $=$, and $<$.

Provability: A *proof system* is a set of axioms and some simple rules of inference. A *standard requirement* is that one can check if a string is a valid proof or not. A *proof* is a sequence of statements, where each statement is an axiom or follows from previous statements by some inference rule. We say that a sentence is *provable in a given proof system* if there is a valid proof (which can be verified for validity).

Desired properties of a proof system:

- **Soundness:** A proof system (for arithmetic) is called *sound* if every provable statement is actually true (over the set $\mathbb{N}$).
- **Completeness:** A proof system (for arithmetic) is *complete* if every true arithmetic sentence is provable in the proof system.

We would like a proof system to be both sound and complete. Then we would have an ability to prove all and only true statements!

Unfortunately, if we want a proof system to reason about arithmetic sentences involving both $+$ and $*$ (with the standard interpretation of these arithmetic operations), then we can't simultaneously have soundness and completeness! This is exactly the statement of Gödel's First Incompleteness Theorem.

We'll give two proofs of the First Incompleteness Theorem: (1) using computability theory, and (2) constructing a self-reference sentence saying "I'm not provable".

2 First proof of Gödel's First Incompleteness Theorem

Theorem 1 (Göedel's First Incompleteness Theorem)**.** *Fix any proof system P that is powerful enough (to reason about + and *). If this proof system P is sound, then it is incomplete.*

In other words, *Truth of arithmetic cannot be captured by any formal proof system (for arithmetic).*

Define the languages

$$Provable = \{\langle \phi \rangle \mid \ \phi \text{ is provable in our proof system } P\}$$

and

$$True = \{\langle \phi \rangle \mid \ \phi \text{ is a true sentence of arithmetic}\}$$

We will argue that $Provable \neq True$ by arguing that exactly one of them is semi-decidable.

Theorem 2. *The language Provable is semi-decidable.*

Proof. Given $\langle \phi \rangle$, try all strings (in lexicographic order), and check if any of them is a valid proof of ϕ. If a valid proof is found, then Accept. □

Theorem 3. *The language True is* not *semi-decidable.*

Proof. By a mapping-reduction from the complement of A_{TM} to $True$. This reduction relies on the following technical and non-trivial lemma.

Lemma 1. *There is an algorithm that, given $\langle M, w \rangle$, produces an arithmetic sentence $\phi_{M,w}$ such that*

$\phi_{M,w}$ is True over $\mathbb{N}$ iff TM M accepts input w.

We'll sketch the proof of this lemma later, but for now, let's assume the lemma and complete our proof that $True$ is not semi-decidable.

Given $\langle M, w \rangle$, our mapping reduction produces the formula $\psi = \neg\phi_{M,w}$, where $\phi_{M,w}$ is as defined in Lemma 1 above.

By definition, we have M does not accept w iff ψ is true. Since the complement of A_{TM} is not semi-decidable, we conclude that True is not semi-decidable. □

Proof sketch of Lemma 1. A TM M accepts w iff there is a sequence of configurations $c_0, c_1, \ldots, c_m$ such that c_0 is an initial configuration of M on w, c_m is an accepting configuration, and each c_i follows from c_{i-1} according to the rules of the TM M. Each configuration can be encoded as a natural number. So we can get a formula with $m+1$ existentially quantified variables (one variable per configuration).

The problem is we don't know the running time of M on w, and so we don't know m. We must come up with a formula of *constant size*, and in particular, with a *fixed* number of variables! To this end, Gödel invented a way to encode *sequences of numbers* with just a pair of numbers: He argued that there is a simple function $\beta(x, y, i)$ (which can be implemented as an arithmetic formula) such that, for every sequence $a_1, a_2, a_3, \ldots, a_m$, there exist numbers A and B so that $\beta(A, B, i) = a_i$ for each $1 \leq i \leq m$. In other words, we can code a sequence of arbitrary length using just two numbers (A and B) and a simple function β to refer to every element of the sequence.

Since we only need two existentially quantified variables (for A and B), we can get a constant-size arithmetic formula (involving the function β) that encodes the statement "there exists a sequence of configurations that is an accepting computation of M on w". □

Note that, if our proof system for arithmetic is sound, then Provable is a subset of True. By the two theorems above, we know that Provable is a *proper* subset of True. That is, there is an arithmetic sentence which is true but not provable. In our second proof of Gödel's First Incompleteness Theorem, we will exhibit such a true yet unprovable sentence for every given proof system.

Finally, note that we showed the language $True$ is not semi-decidable, and so, in particular, it is not decidable. The latter is known as Church's Theorem, which we state below for completeness.

Theorem 4 (Church's Theorem)**.** *There is no algorithm to decide the truth of given arithmetic sentences (involving both $+$ and $*$), i.e., the language True defined above is undecidable.*

2.1 Gödel's function β

Here we provide more details about the function β and its properties. This material is optional.

Define

$$\beta(x, y, i) := x \mod (1 + y(i + 1)).$$

Claim 1. *For every $m \in \mathbb{N}$ and every sequence of m numbers $a_1, \ldots, a_m \in \mathbb{N}$, there exist $A, B \in \mathbb{N}$ such that $\beta(A, B, i) = a_i$ for every $1 \leq i \leq m$.*

Proof. Define $B := (m!) \cdot \max_{1 \leq i \leq m}\{a_i\}$. We want to argue the existence of an A such that $\beta(A, B, i) = a_i$ for all $1 \leq i \leq m$, i.e., that

$$(A \mod b_i) = a_i, \qquad \text{where } b_i := 1 + B(i + 1). \tag{1}$$

Note that by our choice of B, we have $a_i < b_i$ for all $1 \leq i \leq m$, which is a necessary condition for Eq. (1) to hold.

The idea is to use the *Chinese Remainder Theorem*, which says that for any pairwise co-prime natural numbers $c_1, \ldots, c_m$ and any natural numbers $d_1, \ldots, d_m$ where each $d_i < c_i$ (for $1 \leq i \leq m$), there exists a number A such that $(A \mod c_i) = d_i$ for all $1 \leq i \leq m$. To apply the Chinese Remainder Theorem in our case, we need to argue that our numbers $b_1, \ldots, b_m$ (defined in Eq. (1) above) are *pairwise co-prime* (i.e., no prime number p divides both b_i and b_j for some $i \neq j$). We'll argue this next.

Suppose for the sake of contradiction that some prime p exists that divides b_i and b_j for some $i > j$ where $1 \leq i, j \leq m$. It means that $p|1 + B(i + 1)$ and $p|1 + B(j + 1)$. Hence, p also divides the difference $1 + B(i + 1) - (1 + B(j + 1)) = B(i - j)$. Since p is a prime number, we get that either $p|B$ or $p|(i - j)$. But remember that $(m!)|B$ and $1 \leq (i - j) \leq m$, and so $(i - j)|B$. It follows that $p|B$ always.

Next, since $p|B$, we also get that $p|B(i + 1)$. Recall that $p|1 + B(i + 1)$. These two together imply that p divides also their difference $1 + B(i + 1) - B(i + 1) = 1$. We get that $p|1$, and so $p = 1$, but this contradicts the fact that p is a prime number (and hence $p \geq 2$). Thus, we've established that the b_i's are pairwise co-prime. The existence of a requisite number A now follows by the Chinese Remainder Theorem. □

3 Second Proof of Gödel's First Incompleteness Theorem

Here we give another proof of Gödel's First Incompleteness Theorem. The idea of self-reference used in the proof of the Recursion Theorem will be instrumental here as well. In this proof, we exhibit an explicit *true but not provable* sentence. This will later be used to derive Gödel's Second Incompleteness Theorem.

For every given proof system P for arithmetic (powerful enough to talk about $+$ and $*$), we'll construct an arithmetic sentence

G: "G is not provable in P"

That is, G says "I'm not provable".

Assuming we have such a sentence G, we can prove Gödel's Incompleteness Theorem as follows. The sentence G is either true or false. If it is true, then what G says is actually true, i.e., G is not provable. So, we have a true formula G which is not provable.

On the other hand, if G is false, then what it says is false. This means that "G is not provable" is false, and hence, G must be provable. But then our proof system proves the false sentence G. Hence, our proof system is not sound!

The conclusion is: *If our proof system P is sound, then it cannot prove the statement G (which is a true statement in this case).* So, in particular, if P is sound, then it is incomplete.

3.1 "I'm not provable"

Here we show how to construct the sentence G.

The proof relies on two main ideas:

1. **Gödel numbering:** every arithmetic symbol, formula, a sequence of formulas, can be assigned the unique number: $\phi(A) = n$, where n is the Gödel number of formula A.
2. **dual role of numbers:** Numbers can be interpreted as either numbers or encodings of formulas. Hence we can write formulas that talk about (encodings of) other formulas. In particular, we can construct *self-referential* formulas that talk about their own encoding.

We define the following function $proof$:

$$proof(n, m) = \begin{cases} 1 & \text{if } n \text{ is the Gödel number of a proof of the formula with the Gödel number } m \\ 0 & \text{otherwise.} \end{cases}$$

By our assumption, every proof system P is such that we have an algorithm to check validity of proofs. This implies that the function $proof$ defined above is actually a *computable* function. Using ideas of the proof of Lemma 1 from last time (showing that Turing machines can be "simulated" by arithmetic formulas), we get that the function $proof$ has the corresponding arithmetic formula $PROOF(a, b)$ such that $PROOF([n], [m])$ is True iff $proof(n, m) = 1$; here $[n]$ and $[m]$ are numerals (in the proof system) that correspond to natural numbers n and m.

Now, given formula $PROOF$, we can define

$$THM(b) := \exists x \; PROOF(x, b),$$

which means that b has a proof, i.e., that b is (the Gödel number of) a theorem (provable statement) in our proof system P. Observe the formula $\neg THM(b)$ means that b is *not* provable in P.

What is left to do is to somehow make $\neg THM(b)$ refer to its own Gödel number. To this end, we use the following "formula manipulation" function sub:

$$sub(m, n) = j$$

if

- $m = \phi(A(x))$, where x is the only free variable in formula $A(x)$,
- $j = \phi(A([n])$, where $A([n])$ is the new formula obtained from $A(x)$ by substituting the numeral $[n]$ for the free variable x.

In other words, $sub(m, n)$

1. interprets m as the Gödel number of some formula $A(x)$ with a single free variable x,

2. interprets n as a number,

3. constructs a new formula $A([n])$ (obtained from $A(x)$ by plugging in n for x),

4. outputs the Gödel number j of the constructed sentence $A([n])$.

The function *sub* is easy to compute, and hence (again along the lines of Lemma 1 from last time) it can also be represented by an arithmetic formula. With some abuse of notation, let's denote the arithmetic formula computing *sub* by Sub and assume that the $Sub([m],[n]) = [j]$ iff $sub(m,n) = j$. (In reality, we need to define a formula $SUB(x,y,z)$ such that $SUB([m],[n],[j])$ is true iff $sub(m,n) = j$.)

Now we can construct a formula that says "I am not provable". First, consider the formula

$$A(x) := \neg THM(Sub(x,x))$$

with a single free variable x. Let's assume that the Gödel-number of $A(x)$ is n_0. Define the new formula (without free variables):

$$G := A([n_0]),$$

i.e., we have (by the definition of $A(x)$) that

$$G \equiv \neg THM(Sub([n_0],[n_0])).$$

Observe that the Gödel number of G is $sub(n_0,n_0)$ (because G is obtained from $A(x)$ with the Gödel number n_0 by substituting $[n_0]$ for the free variable x in $A(x)$).

On the other hand, by definition, the sentence G says

"The formula with the Gödel number $sub(n_0,n_0)$ is not provable in P".

Hence, G says "G is not provable in P", which is equivalent to "I am not provable in P".