Lecture 11:
Gödel's Second Incompleteness Theorem, and Tarski's Theorem

Valentine Kabanets

October 27, 2016

1 Gödel's Second Incompleteness Theorem

1.1 Consistency

We say that a proof system P is *consistent* if P does not prove both A and $\neg A$ for some sentence A. That is, a consistent proof system cannot derive a contradiction $A \wedge \neg A$.

In the case of a proof system P for arithmetic, we get that P is consistent iff P does not prove the sentence "$1 = 2$" (since $1 \neq 2$ can be derived in P by the usual axioms (of Peano arithmetic) for the natural numbers).

While *soundness* defined earlier is a semantic notion (referring to the truth of arithmetic sentences), *consistency* is a syntactic notion (referring to the syntactic provability of a particular sentence $1 = 2$ within a given proof system). In particular, it is possible to write an arithmetic formula expressing the consistency of a proof system P, denoted $Cons_P$, as follows:

$Cons_P$: "the sentence "$1 = 2$" is not provable in P",

where we use the fact established earlier (in our second proof of Gödel's First Incompleteness Theorem) that "provability in P" is expressible by an arithmetic formula.

Consistency of a proof system is a minimal requirement we may want to impose. Even if the system is not sound, at least we want it to be free of internal inconsistencies. Can a consistent proof system be complete? No! In fact, as Gödel showed, a consistent proof system is unable to prove its own consistency!

1.2 Incompleteness Theorem

Here we'll prove the following.

Theorem 1 (Gödel's Second Incompleteness Theorem). *Fix any proof system P that is powerful enough (to reason about + and *, plus satisfying some provability conditions). If P is consistent, then the consistency of P (i.e., the statement $Cons_P$) cannot be proved by P.*

Proof sketch. First, we observe that our second proof of Gödel's First Incompleteness Theorem (given last time) proves something more:

Claim 1. *Assuming that a proof system P is* consistent *(rather than* sound*), we conclude that the sentence G (saying "I'm not provable in P") is not provable in P (i.e., G is actually true).*

Proof of Claim. Let g be the Gödel number of the sentence G. Recall that G is logically equivalent to the statement

$$\neg\exists x\ PROOF(x, [g]),$$

where $PROOF(x, y)$ means that x is a Gödel number of a proof of the statement Y whose Gödel number is y.

We will argue by contradiction. We assume that G is false, and show that P must then be inconsistent.

So suppose that G is false. This means that P proves G. Hence, there is a natural number m_0 which is the Gödel number of a proof of G in P. Then the sentence

$$PROOF([m_0], [g])$$

is true. The truth of this sentence can be proved within P itself![1] This implies that P also proves the formula

$$\exists x\ PROOF(x, [g]).$$

The last formula is equivalent to $\neg G$. So we get that P proves $\neg G$. But we assumed earlier that P proves G. These two statements together imply that P proves the contradictory statement $G \wedge \neg G$, and hence, P proves $1 = 2$. Thus, P is inconsistent!

Therefore, we conclude that if P is consistent then G is true, i.e., that the following implication is a true statement:

$$Cons_P \Rightarrow G \tag{1}$$

This concludes the proof of the claim. □

The proof of Claim 1 (given above) can in fact be *formalized within the proof system P* (for any proof system P for arithmetic such that P is powerful enough to talk about $+$ and $*$, plus satisfying some additional provability conditions). That is, P proves the implication in Eq. (1).

Now, if we suppose that P proves its own consistency, i.e., if P proves $Cons_P$, we get from Eq. (1) (by *modus ponens*) that P also proves G. Moreover, we get (by the provability assumptions on the proof system P we made) that P proves that P proves G. In particular, since the statement "P proves G" is equivalent to $\neg G$, we get that P proves $\neg G$.

Thus, we get that P proves G, and that P proves $\neg G$. Hence, P is inconsistent. We conclude that a consistent proof system P cannot prove its own consistency $Cons_P$. □

1.3 Consequences

So to prove the consistency of your system, you need to go to a larger system. But then to prove the consistency of the larger system, you need to go to an even larger system, and so on.

The upshot is that one just has to take on faith that standard proof systems for arithmetic (such as Peano Arithmetic) are consistent. The consistency of Peano Arithmetic can be proved inside the standard system for Set Theory, but then we cannot prove the consistency of Set Theory inside Set Theory. So at some point, one just needs to take on faith that things are consistent, and keep their fingers crossed.

[1] This is one of the provability assumptions that we must make about the proof system P: namely, if for two natural numbers a and b, the formula $PROOF([a], [b])$ is actually true, then the system P proves the formula $PROOF([a], [b])$. This can be thought of as some "restricted completeness" of P: all true formulas of the form $PROOF([a], [b])$ are in fact provable in P.

2 Analogies between Computability and Logic

In Computability, we first gave a non-constructive argument that undecidable problems exist. In Logic, we first argued that there exist true but unprovable arithmetic statements.

In Computability, we then showed that an artificial language D (obtained by diagonalization) is an example of a language that is not even semi-decidable. In Logic, we constructed an artificial sentence G (saying "I'm not provable"), which is an example of a true and unprovable statement.

In Computability, we finally argued that natural problems, such as the Halting Problem, are undecidable, via a reduction to the language D. In Logic, we proved that a natural statement, such as the Consistency of a Proof System P for Arithmetic, is not provable within the proof system P itself if P is indeed consistent, where our argument used the previously constructed sentence G.

Next, in Computability, we showed the Recursion Theorem, saying that one can construct Turing machines that apply any given computable function to their own code. Analogously, we show next a similar statement for arithmetic formulas.

3 The recursion theorem for arithmetic formulas

3.1 Recursion Theorem

Using the ideas from the second proof of Gödel's First Incompleteness Theorem, we get the following version of the "recursion theorem" for arithmetic formulas. Informally, for any formula $A(x)$, there is a formula B that applies A to its own Gödel number (i.e., to itself).

Theorem 2 ("Recursion Theorem for Arithmetic Formulas")**.** *Let $A(x)$ be any arithmetic formula of a single free variable x. Then there exists an arithmetic sentence B such that*

$$B \equiv A([n_0]),$$

where n_0 is the Gödel number of B.

Proof. Using the *sub* function defined above, we define

$$C(x) := A(Sub(x, x)).$$

Let a be the Gödel number of the formula $C(x)$. Define $B := C([a])$, i.e.,

$$B \equiv A(Sub([a], [a]).$$

Observe that the Gödel number of B is $n_0 := sub(a, a)$, and $B \equiv A([n_0])$. □

3.2 Tarski's Theorem

As an application of the recursion theorem for arithmetic formulas, we can prove the following result of Tarski saying that *truth of arithmetic statements cannot be encoded with an arithmetic formula.*

Theorem 3 (Tarski's Theorem)**.** *There is no arithmetic formula $Truth(x)$ such that, for every n, $Truth([n])$ holds iff n is the Gödel number of a true sentence.*

Proof. Suppose $Truth(x)$ can be expressed as an arithmetic formula. Apply the Recursion Theorem (Theorem 2) to the formula $\neg Truth(x)$, getting the sentence saying "I'm false", which is a contradiction.

In more detail, consider the following formula: $\neg Truth(Sub(x, x))$, where $Sub()$ is, as before, the substitution formula. Let n_0 be the Gödel number of this formula. Consider the sentence

$$\neg Truth(Sub([n_0], [n_0])).$$

It is the formula with Gödel number $sub(n_0, n_0)$. At the same time, this formula says that the formula with the Gödel number $sub(n_0, n_0)$ is false. So this formula says "I'm false", which is a self-contradictory statement. This contradiction shows that $Truth(x)$ cannot be expressed as an arithmetic formula. □

We note that Tarski's theorem above generalizes Church's Theorem (proved earlier) saying that Peano arithmetic is not decidable.

The reason is: Suppose Peano arithmetic were decidable by some decider TM M such that, for every formula π, $M(\langle\pi\rangle)$ accepts iff π is true. Then we can define the arithmetic formula

$$\psi(x) := \phi_{M,x},$$

where the formula $\phi_{M,w}$ comes from Lemma 1 in Lecture 10. We get that $\psi(x)$ is true iff M accepts x, i.e., iff x is an encoding of a true arithmetic sentence. Thus, we would get $\psi(x) \equiv Truth(x)$. (In fact, the given argument actually shows that to get the formula $Truth(x)$, it suffices to assume that Peano arithmetic is semi-decidable.)

Thus, in particular, Tarski's Theorem implies Church's Theorem.

3.3 "I'm provable"

The material in this subsection is *optional*.

Consider a proof system P (as in the setting of Gödel's Theorem), and define an arithmetic sentence H to be "H is provable in P". That is, H says "I'm provable". Such a sentence is possible to construct thanks to the Recursion Theorem (Theorem 2), using the arithmetic formula $THM(x)$ (saying "x is provable") defined earlier.

This sentence H was considered by Henkin, who asked whether H is true. We'll show that indeed H is true! So H is indeed true and provable!

Theorem 4. *The sentence H saying "I'm provable in P" is indeed true.*

Proof. Let's use the notation "$[]A$" to mean "A is provable in P".

We'll need the following lemma.

Lemma 1 (Löb's lemma)**.** *If P proves the implication $[]A \to A$, then P proves A.*

Kreisel's proof of the Lemma. Define the new proof system P' to be P plus the axiom $\neg A$.

Clearly, P' also proves $[]A \to A$. But now, since P' also proves $\neg A$, we conclude that P' proves $\neg[]A$. That is, P' proves that "the statement A is *not* provable in P".

The latter implies that P' proves that "A is not provable in P'". Indeed, if A is provable in P, then A is also provable in P' (by simply ignoring the axiom $\neg A$). Conversely, if A is provable in P', then P proves $A \vee A$, which is A (by basic logic). So, provability of A in P and in P' are equivalent.

Finally, consider the statement "A is not provable in P'". If P' is inconsistent (i.e., P' can derive a contradiction), then P' proves everything, including A. The fact that A is not provable in P' thus implies that P' is consistent. Moreover, P' itself proves that fact, i.e., P' proves its own consistency! But, by Gödel's Second Incompleteness Theorem (in the contrapositive form), we conclude that such P' is inconsistent. Hence, $P, \neg A$ derives a contradiction. Hence, P proves A. $\square$

Now consider our H. It is equivalent to $[]H$. Hence, in P, we can prove the implication "$[]H \to H$". Using Löb's lemma, we conclude that P proves H, which means that H is true. $\square$

4 Decidability of Presburger's Arithmetic

We've shown that a proof system for arithmetic that talks about both $+$ and $*$ cannot be both sound and complete. In contrast, there is a proof system for arithmetic that talks about $+$ only (and not $*$), which is called *Presburger Arithmetic*, that is both sound and complete! A proof can be found in our textbook (Sec. 6.2), and is a beautiful application of the Finite Automata Theory to Logic!

5 Summary of Logic

- No proof system for arithmetic (including $+$ and $*$) can be both sound and complete [Gödel's First Incompleteness Theorem].
- The sentence G="I'm not provable" is an example of a true but unprovable sentence [Gödel's sentence G].
- In fact, if such a proof system is consistent, then it cannot prove its own consistency [Gödel's Second Incompleteness Theorem].
- Truth of arithmetic sentences cannot be encoded by an arithmetic formula [Tarski's Theorem].
- In particular, the truth of Peano Arithmetic (involving both $+$ and $*$) is undecidable [Church's Theorem].
- Similarly to Computability Theory (diagonalization), the proofs of Gödel's Theorems rely on *self-reference* (constructing a sentence G saying "I'm not provable").
- In contrast, arithmetic for $+$ only (Presburger arithmetic) is decidable, and also admits a proof system which is both sound and complete [can be argued using Finite Automata Theory].
- The sentence H="I'm provable" is in fact true [Henkin's sentence H].