185.A09 Advanced Mathematical Logic

www.volny.cz/behounek/logic/teaching/mathlog13

Libor Běhounek, behounek@cs.cas.cz

Lecture #7, January 21, 2014

Undecidability and incompleteness of arithmetic

Not all theories are decidable. Examples of undecidable theories include Robinson, Peano, and True Arithmetic. True Arithmetic is not even recursively enumerable; consequently, PA and Q are also incomplete (PA is recursive, so Thm(PA) *is* recursively enumerable; thus $\mathrm{Th}(N) \setminus \mathrm{Thm}(\mathrm{PA}) \neq \emptyset$.)

The above claims constitute (a particular form of) *Gödel's Incompleteness Theorem.* The main idea of the proof, due to Gödel, consists in formalizing (or encoding) the syntax of arithmetical theories in (Peano or Robinson) arithmetic. In this way, arithmetical formulae become capable of referring to (encoded) arithmetical formulae and their provability, and a diagonal argument showing the incompleteness and undecidability can be made.

A full proof of Incompletness Theorems exceeds the scope of this course; therefore we just sketch the main ideas underlying the proof (for more details see standard textbooks on classical mathematical logic).

Arithmetical characterization of recursivity

In order to capture the notion of *recursivity* in the framework of arithmetization, we need a syntactic criterion for decidability and recursive enumerability of sets delimited by arithmetical formulae:

Definition. An arithmetical formula φ is:

- *Bounded* if every quantification in φ is bounded, i.e., is of the form $(\forall x \leq y)\psi$ or $(\exists x \leq y)\psi$ (which abbreviates $(\forall x)(x \leq y \to \psi)$ and $(\exists x)(x \leq y \wedge \psi)$, resp.).
- A Σ_1-*formula* if it is of the form $(\exists x_1, \ldots, x_k)\psi$, where ψ is a bounded formula.
- A Δ_1-*formula* if both φ and $\neg\varphi$ are Σ_1-formulae.
- A bounded/Σ_1/Δ_1 *in a theory* T if there is a bounded/Σ_1-/Δ_1-formula φ' such that $T \vdash \varphi \leftrightarrow \varphi'$.

Example. The formulae $\mathrm{Odd}(x) \equiv_{\mathrm{df}} \neg(\exists y \leq x)(x = y + y)$ and $x \mid y \equiv_{\mathrm{df}} (\exists z \leq y)(x \cdot z = y)$ are bounded.

Observation. In any theory T, bounded and Δ_1-formulae are closed under $\wedge$, $\vee$, $\neg$, and bounded quantification, and Σ_1-formulae are closed under $\wedge$, $\vee$, and $\exists$.

It can furthermore be shown that the class of Δ_1-formulae is additionally closed under quantification bounded by defined functional terms (rather than just variables, as required by the definition of bounded quantification).

Observation. If $\varphi(x_1, \ldots, x_k)$ is a bounded/Σ_1-/Δ_1-formula, then the set delimited by φ, i.e., $\{\langle n_1, \ldots, n_k\rangle \in N^k \mid N \models \varphi(n_1, \ldots, n_k)\}$, is primitive recursive / recursively enumerable / recursive. (*Hint:* Bounded: evaluate bounded quantifications as disjunctions of instances from 0 up to the bound. Σ_1: evaluate the existential quantifier by a **while**-cycle from 0 upwards. Δ_1: by Post's theorem.)

It can in fact be shown that the classes of sets delimited by Δ_1- / Σ_1-formulae *coincide* with the classes of recursive / recursively enumerable sets in N.

Arithmetization of syntax

The syntax of first-order logic has been formulated by means of finite set theory: for instance, a formula is defined as a finite sequence of symbols; a proof as a finite sequence of formulae; etc. To represent the syntax within arithmetic, we need first to represent the basic set-theoretic notions of *pair, finite set,* and *finite sequence* in arithmetic. There are many possible encodings suitable for our purposes; we will choose one which is close to Gödel's original proof and is easily expressible in Q. We will take care to define our encoding by Δ_1- (sometimes even bounded) or at worst Σ_1-formulae, to ensure their algorithmic tractability.

1. *Pairs.* Observe that the function $f \colon \langle m, n\rangle \mapsto \frac{1}{2}(m+n)(m+n+1)+m$ is a bijection $N^2 \to N$. It can be formalized in arithmetical language by the defined predicate $\mathrm{Pair}(x, y, z) \equiv_{\mathrm{df}} \overline{2} \cdot z = (x+y) \cdot (x+y+\overline{1}) + \overline{2} \cdot x$, expressing that "the number z encodes the pair $\langle x, y\rangle$". Observe that Pair is an open (so bounded) formula. Moreover, if $x, y > 1$, the code z is larger than both x, y (a property important later for obtaining bounds on quantifiers and for induction on the length of formalized formulae). Importantly, the requisite properties of Pair are *provable* in PA (most of them already in Q): e.g., $\mathrm{PA} \vdash (\forall x, y)(\exists! z)\, \mathrm{Pair}(x, y, z)$, the bijectivity of Pair, etc. (these properties are thus ensured if PA is our metamathematics).

2. *Finite sets.* Our encoding of finite sets of numbers by a single number will make use of the defined predicate $\mathrm{Lcm}(x, y)$ expressing "y is the least common multiple of all numbers $0 < i \leq x$"; it can be defined by a bounded formula, using the (bounded) predicate $x \mid y$ (exercise). Again, PA 'knows' its requisite properties (e.g., $\mathrm{PA} \vdash (\forall x)(\exists! y)\, \mathrm{Lcm}(x, y)$, $\mathrm{PA} \vdash \mathrm{Lcm}(x, y) \to x \leq y$, etc.).

   The empty set $\emptyset$ is encoded by the number 0. For non-empty sets, the encoding of $A = \{n_1, \ldots, n_k\} \subseteq N$ is done as follows: let $n = \max(n_1, \ldots, n_k)$; take m such that $\mathrm{Lcm}(n, m)$; let $p = \prod_{i=1}^{k}\bigl(1 + (1+n_i) \cdot m\bigr)$; the code of A is then the (code of the) pair $\langle m, p\rangle$.

   The decoding of $\langle m, p\rangle \neq 0$ is done by taking the largest n such that $\mathrm{Lcm}(n, m)$; the set A then consists of all $\ell \leq n$ such that $1 + (1+\ell) \cdot m \mid p$.

   Based on this description, it should be clear how to formally define the arithmetical predicates $\mathrm{Set}(x)$ of being a code of a set (left as an exercise) and $x \in y$ ("x is an element of the set encoded by y"):

$$x \in y \equiv_{\mathrm{df}} \mathrm{Set}(y) \wedge y \neq 0 \wedge (\exists u, v \leq y)\bigl(\mathrm{Pair}(u, v, y) \wedge \overline{1} + (\overline{1} + y) \cdot u \mid v\bigr)$$

Observe: $\in$ (as well as Set) is defined by a bounded formula, and the code of a set is larger than all of its elements (important later for bounds on quantifiers and inductive proofs). Again, PA can prove all requisite properties of Set and $\in$ (including, e.g., the extensionality of sets).

3. *Finite sequences* are encoded straightforwardly as sets of pairs. Predicates for common manipulations with sequences can be defined by arithmetical formulae, in particular: $\mathrm{Seq}(x)$ "x is a code of a sequence", $\mathrm{Len}(x,y)$ "y is the length of the sequence x", and $\mathrm{Cat}(x,y,z)$ "z is the concatenation of the sequences y and z". All of these predicates are bounded and PA 'knows' their requisite properties.

With this set-theoretic machinery in place, we can proceed to encoding the syntax of first-order logic:

4. *Logical symbols:* Let us encode variables $x_1, x_2, \ldots$ by odd numbers (being a variable is thus a bounded condition) and the symbols $(,), \rightarrow, \neg, \forall, =, s, +, \cdot$ respectively by the numbers $0, 2, 4, \ldots, 16$. (In languages with further function or predicate symbols use further even numbers.)

5. *Terms* can suitably (and equivalently to our definition from Handout 1) be defined as the last elements of *term formation sequences,* i.e., sequences whose each element is either a variable, or arises from applying a function symbol to preceding elements of the sequence. (For example, a term formation sequence for the term $(s(0)+s(0))\cdot s(s(0))$ is: $0, s(0), s(s(0)), s(0)+s(0), (s(0)+s(0))\cdot s(s(0))$.) The syntactic criterion for being a term formation sequence can easily be translated into an arithmetic condition on the codes of sequences (exercise). Moreover, since obviously the length of a minimal term formation sequence as well as the length of each of its elements must be smaller than the length of the term itself, an upper bound on the code of a minimal term formation sequence for a given term t can be given as a definable function of (the code of) t. Consequently (by the earlier mentioned fact that Δ_1 is closed under quantification bounded by definable functions of variables) the formula defining the predicate $\mathrm{TFS}(x)$ "x is a term formation sequence", and thus also the predicate $\mathrm{Term}(x)$ of being a term, is Δ_1. The encoding is clearly injective, i.e., different terms are assigned different codes. (Again, these facts are provable in PA.)

6. *Formulae* can be defined by a Δ_1-condition analogously as terms (by means of formula formation sequences). Again, the encoding is injective and its requisite properties are provable in PA. The code of a formula φ (in a given encoding) is called the *Gödel number* of φ and denoted by $\overline{\varphi}$. (The same notation is employed also for parts of formulae, including terms and logical symbols: e.g., $\overline{\rightarrow}$, $\overline{x}$, etc.)

7. *Theories* are (possibly infinite) sets of formulae. We are mainly interested in recursive theories, as only these are algorithmically tractable. The recursive set of axioms of a given theory T can be delimited by a Δ_1-formula $\tau(x)$; we say that τ is the *formalization* of the theory T.

   Example: The formalization $q(x)$ of Robinson arithmetic Q is the disjunction of 9 clauses comparing x to the Gödel codes of the 9 axioms of Q:

$$q(x) \equiv_{\mathrm{df}} x = \overline{s(x) = s(y) \rightarrow x = y} \vee x = \overline{\neg(s(x) = 0)} \vee \text{etc.}$$

   (As an advanced exercise, put down the formula $\pi(x)$ formalizing PA.)

8. *Provability in a recursive theory.* Given a recursive theory T formalized by a Δ_1-formula τ, it is an easy (though tedious) exercise to put down a Δ_1-formula $\mathrm{Prf}_\tau(x, y)$ expressing the condition that "x encodes a proof of a formula encoded by y". Notice that τ is a *subformula* of Prf_τ (corresponding to the clause "… or the formula is an axiom of T …" in the definition of proof) rather than its argument. (Thus, it is "hardwired" in $\mathrm{Prf}_\tau(x, y)$: for each τ we have a *different* formula $\mathrm{Prf}_\tau(x, y)$, though algorithmically constructible from τ.)

   The property of being provable in T is then formalized by the predicate $\mathrm{Pr}_\tau(y) \equiv_{\mathrm{df}} (\exists x)\,\mathrm{Prf}_\tau(x, y)$, a Σ_1-formula. (We are unable to give a bound for x: short theorems can have long proofs.) Again, PA proves many properties of Pr_τ, e.g., $\mathrm{PA} \vdash \mathrm{Pr}_\tau(\overline{\varphi}) \wedge \mathrm{Pr}_\tau(\overline{\varphi \to \psi}) \to \mathrm{Pr}_\tau(\overline{\psi})$, etc.

   Finally, we can define formal consistency of T as the sentence $\mathrm{Con}_\tau \equiv_{\mathrm{df}} \neg\,\mathrm{Pr}_\tau(\overline{\bot})$. (Notice that it is $\neg\,\mathrm{Con}_\tau$ which is a Σ_1-formula, rather than Con_τ itself: indeed, successive generation of all proofs in T will eventually yield a proof of a contradiction if T is inconsistent, while no step of the generation can confirm its consistency.)

The definitions ensure the intended meaning of these defined predicates in the standard model N of arithmetic (e.g., $N \models \mathrm{Pr}_\tau(\overline{\varphi})$ iff φ is provable in T). However, in non-standard models of arithmetic, the predicates are satisfied as well by some non-standard ("infinite") numbers, which do not encode any formula or proof of finite length. Thus in general we have to distinguish between *formal* provability, *formal* consistency, etc. (defined by the above predicates in PA) and our usual metamathematical notions of provability, consistency, etc.

Gödel's Incompleteness Theorems

Several variants of Incompleteness Theorems (with assumptions of varying strengths) can be proved; for simplicity, we will formulate the theorems just for PA, even though many of them already work for Q. We skip the numerous lemmata needed for the proof (in particular, the provability in PA of requisite properties of the Gödel encoding and formal provability, and the representability of all recursive functions in PA).

With the syntax of PA represented within PA, arithmetical formulae with free variables can be applied to Gödel numbers of arithmetical formulae, and so express various (formalized) properties of arithmetical formulae (such as their provability in recursive theories). In particular, they apply to *their own* Gödel codes, and so are subject to various diagonal arguments. For any sound (i.e., $N \models T$) recursive theory T extending PA, a particular (rather involved) diagonalization argument produces an (explicitly constructed) sentence which "claims its own unprovability" (or more rigorously, which is equivalent to its own formalized unprovability, i.e., $T \vdash \varphi \leftrightarrow \neg\,\mathrm{Pr}_\tau(\overline{\varphi})$). Such a sentence has to be true in N, and therefore unprovable in T (if it were false in N, by $\mathrm{PA} \vdash \neg\varphi \leftrightarrow \mathrm{Pr}_\tau(\overline{\varphi})$ it would provable in T, a contradiction with the assumed soundness of T). This yields *Gödel's First Incompleteness Theorem:* Any sound recursive theory T extending PA is incomplete. (A further argument shows that any such T is also undecidable.)

A crucial step in the proof of the Incompleteness Theorem is the *Autoreference* (or *Diagonal*) *Lemma:* For any arithmetical formula $\psi(x)$ with a single free variable x there is a sentence φ such that $\mathrm{PA} \vdash \varphi \leftrightarrow \psi(\overline{\varphi})$. (Such a 'fixed-point' sentence φ can be algorithmically constructed by a diagonalization trick: given ψ, the sentence φ is defined as the formula $\chi(\overline{\chi})$ for a suitable formula χ algorithmically constructed from ψ, and $\mathrm{PA} \vdash \chi(\overline{\chi}) \leftrightarrow \psi(\overline{\chi(\overline{\chi})})$ then follows by PA-provable properties of the Gödel encoding.) Gödel's Incompleteness Theorem is then obtained by taking the formula $\neg \mathrm{Pr}_T(x)$ for $\psi(x)$, which yields $\mathrm{PA} \vdash \varphi \leftrightarrow \neg \mathrm{Pr}_T(\overline{\varphi})$.

A corollary to Gödel's First Incompleteness Theorem is an observation (following easily from the First Incompleteness Theorem by PA-provable properties of provability) called *Gödel's Second Incompleteness Theorem:* If T is a consistent theory containing PA, then $T \nvdash \mathrm{Con}_T$ (i.e., T does not prove its own formal consistency).

In particular, the Second Incompleteness Theorem shows that the consistency of set theory (say, ZF) cannot be proved in sound recursive theories weaker than ZF. This refutes the possibility of *Hilbert's Program* of finitistic justification of infinitistic methods in mathematics (under most plausible interpretations of "finitistic").

Besides their implications for the philosophy of mathematics, Gödel's Incompleteness Theorems are useful for showing some negative metamathematical results, such as the following:

- Classical *second-order logic* (i.e., logic with variables and quantifiers not only for elements, but also subsets of the domain of discourse) cannot be recursively axiomatized, since it interprets True Arithmetic. (Recursive axiomatization is thus prevented by the First Incompleteness Theorem.)
- Zermelo–Fraenkel set theory ZF is not finitely axiomatizable. (If ZF were finitely axiomatized, then by the so-called reflection principle ZF would prove the existence of a model of ZF, and so its own consistency—a contradiction with the Second Incompleteness Theorem.)
- The existence of the *satisfaction relation* (i.e., the relation formalizing satisfaction of formulae under evaluations) is not provable in set theory. (Formal consistency of the theory is derivable from the existence of the satisfaction relation—a contradiction with the Second Incompleteness Theorem).

However, it should be stressed that many philosophical interpretations of the Incompleteness Theorems are misguided. (For one reason, notice that the incompleteness only regards truly infinite sets of natural numbers: arithmetic up to any finite number, however huge, is complete and decidable, being a theory of a finite model.) A comprehensive book on the misinterpretations of Gödel's theorems is Torkel Franzén: *Gödel's Theorem: An Incomplete Guide to its Use and Abuse.* A.K. Peters, 2005.

Acknowledgment: Two Czech textbooks (A. Sochor: *Klasická matematická logika,* V. Švejdar: *Logika: neúplnost, složitost, nutnost*) have been used in preparation of these handouts.