Gödel's Incompleteness Theorem

notes from a talk by Enrico Martino

March 22, 2004

1 Peano Arithmetic - Syntax and Semantics

We work with the language of First Order Predicate Calculus with identity. The primitive symbols of Peano Arithmetic are: the individual constant "0", the symbol "S" for the successor function and the symbols "+" and "$\cdot$" for the operation of addition and multiplication. The terms (expressions that are to denote numbers) are defined by the following grammar (where x ranges over an infinite list of individual variables):

$$t \ := \ 0 \mid x \mid St \mid t_1 + t_2 \mid t_1 \ \cdot \ t_2$$

In particular any natural number n can be denoted by a term of the form $S \ldots S0$ (n occurrences of "S" before "0"), with $n \geq 0$. These terms are called *numerals*; if n is a natural number, then $\underline{n}$ is the numeral denoting n.

Atomic formulas are equalities between terms: $s = t$. Complex formulas are built up from these using connectives and quantifiers. A formula without free variables will be called a *sentence* (or *proposition*). A formula containing free variables will be called *propositional function*: these formulas express arithmetic properties and relations).

Peano Arithmetic can be developed as a deductive system from the following axioms (*Peano's axioms*):

Ax 1. $\forall x.\neg(Sx = 0)$;

Ax 2. $\forall x \forall y.(Sx = Sy \rightarrow x = y)$;

Ax 3. $\forall x.x + 0 = x$;

Ax 4. $\forall x \forall y.x + Sy = S(x + y)$;

Ax 5. $\forall x.x \cdot 0 = 0$;

Ax 6. $\forall x \forall y. x \cdot Sy = (x \cdot y) + y$;

Ax 7. Axiom Scheme of Induction: for every propositional function $A(x)$

$$(A(0) \wedge (\forall x. A(x) \rightarrow A(Sx))) \rightarrow (\forall y. A(y)).$$

The induction principle is expressed here as an *axiom scheme*, rather than a single axiom: such a scheme includes an infinity of axioms, one for each propositional function. Intuitively, this means that we assume the principle of induction for all *arithmetic properties*, i.e., for all properties expressible in the formal language of arithmetic.

An *interpretation* in the language of arithmetic is obtained by fixing a universe of discourse U and by assigning an individual of U to the individual constant 0, a unary function (from U into U) to the symbol "S", and two binary functions to the symbols "+" and "·". A *model* of arithmetic is an interpretation which verifies Peano's axioms. In particular, these are evidently verified by the intended model, called also the *standard model*: the set N of natural numbers, with the usual interpretations of the arithmetic symbols "S", "+" and "·". However, we shall see that there are also other models.

Two models M, M' (of arithmetic) are called *isomorphic* if there exists a bijection $f : M \rightarrow M'$ which preserves the interpretations of the arithmetic symbols. Namely, f maps the zero of M to the zero of M', the successor of an element $m \in M$ the successor of $f(m) \in N'$, and so on. Formally,

$$f(0_M) = 0_{M'}; \quad f(S_M(a)) = S_{M'}(f(a));$$
$$f(a +_M b) = f(a) +_{M'} f(b) \quad f(a \cdot_M b) = f(a) \cdot_{M'} f(b)$$

Such a function shall be called an *isomorphism* between M and M'.

Two isomorphic models can be regarded as "essentially the same model", in the sense that, although they may consist of different individuals, there s abijection between the two sets of individuals and moreover each individual behaves in the same way as its correspondent with respect to the arithmetic operations.

An axiomatic theory is said to be *categorical* if all its models are isomorphic, i.e., if it has essentially only one model. Thus a categorical axiom system identifies a well-specified mathematical structure.

Since Peano Axioms have been conceived to characterize the structure N of the natural numbers with the familiar operations, i.e., the standard model, it is natural to ask whether such an axiomatic system is categorical and hence whether if it succeeds in determining the intended model. We shall see that the answer is negative.

1.1 Non-standard models

We show that there are models of Peano Arithmetic non-isomorphic to the standard model. We use the following theorem of first order logic:

Theorem. (*Model existence*) *Every consistent set of formulas has a model.*

This theorem is a corollary of the following fundamental theorem:
Completeness Theorem. *Let A be a formula of a first-order language $\mathcal{L}$. If A is true in every interpretation M of $\mathcal{L}$, then A can be deduced in (one of the familiar) deductive systems for predicate logic.*

Indeed, if a set Γ of formulas does not have a model, then $A \wedge \neg A$ is true in every model of Γ (trivially, since because there is no model of Γ). By the completeness theorem, there is a formal derivation of $A \wedge \neg A$ from Γ, and this means that Γ is not consistent, a contradiction.

Indeed, from the completeness theorem we can derive a more powerful corollary:

Compactness Principle: *If every finite subset of a set Γ of formulas has a model, then Γ has a model.*

Indeed, if Γ did not have a model we could derive a contradiction $A \wedge \neg A$ from a *finite subset* Γ_0 of Γ: indeed, a derivation of any formula is a finite entity, which contains only a finite number of formulas in Γ. But then there can be no model of Γ_0, because $\neg \bigwedge(\Gamma)$ is *valid*, i.e., true in every models. Here we have used also the converse of the Completeness theorem:

Soundness Theorem: *Let A be a formula of a first-order language $\mathcal{L}$. If A can be deduced in (one of the familiar) deductive systems for predicate logic, then A is true in every interpretation M of $\mathcal{L}$.*

In the language $\mathcal{L}$ of arithmetic we can define an order relation:

$$a > b =_{df} \exists x.(\neg x = 0 \wedge a = b + x).$$

Let $\mathcal{L}'$ be the language obtained from $\mathcal{L}$ by adding a new individual constant c. Consider the following infinite sequence H of formulas of $\mathcal{L}'$:

$$H = \{c > \underline{0}, c > \underline{1}, \ldots, c > \underline{n}, c > \underline{n+1}, \ldots\}$$

Let $\mathbf{V}$ be the set of formulas of $\mathcal{L}$ which are true in the standard model. We claim that the set $\mathbf{V} \cup H$ has a model. Indeed, let $H_n = \{c > \underline{0}, c > \underline{1}, \ldots, c > \underline{n}\}$; then for every n, we have that $\mathbf{V} \cup H_n$ has a model, because we may interpret c as $n+1$. By compactness, $\mathbf{V} \cup H$ has a model M, which is called a *non-standard* model. In M there are all the *standard numbers*, namely, the interpretations of the numerals $\underline{n}$. But there are also *non-standard numbers*, such as c. Notice that the presence of c implies the existence in M of *infinitely many* non-standard numbers greater than c: indeed, the formula $\forall x.\exists y.y > x$ is true in the standard model N, and so it must be true also in M. Similarly, $\forall x.\neg x = 0 \rightarrow \exists y.x > y$ is true in the standard model N, and therefore in M, and since $\neg c = 0$, $\neg(c-1) = 0$, $\ldots$ are all true, this implies that there are infinitely many non-standard numbers less than c. Now it is obvious that M cannot be isomorphic to N. If $f : N \rightarrow M$ is an injective function which preserves the successor function, then it can only send an element of N into the standard part of M, i.e., the non-standard numbers cannot be in the codomain of f from N to M, hence f is not surjective and thus cannot be an isomorphism.

We have shown that the axiom system of Peano Arithmetic is not categorical. We cannot obtain categoricity by suitably expanding this axiom system by taking other formulas of the language $\mathcal{L}$ as new axioms. Indeed let A be any set of axioms expressible in $\mathcal{L}$: if $\mathbf{A}$ describes correctly the intended model N, then $\mathbf{A}$ must be a subset of the set $\mathbf{V}$ of formulas of $calL$ which are true in N; but then $\mathbf{A}$ cannot be categorical, because $\mathbf{V}$ itself is not categorical. In this sense we can say that Peano Arithmetic is *essentially* non-categorical.

Let us regard M as an interpretation of the language $\mathcal{L}$ (thus, not including the symbol c for a non-standard number): then the same set of formulas in the language of $\mathcal{L}$ is true in M *and* in N. This means that the language $\mathcal{L}$ of arithmetic cannot distinguish between N and M. Two interpretations of a given language $\mathcal{L}$ which satisfy the same set of formulas are said *elementary equivalent*. Thus we can say that M and N are not isomorphic, but they are elementary equivalent.

However the fact that Peano arithmetic is non-categorical does not rule out the possibility that there could be a *complete axiomatization*, i.e., a system of axioms A such that every formula true in N is deducible from A. The famous *incompleteness theorem* by Gödel rules out this possibility, as we shall see now.

1.2 Coding and arithmetic representability

The expressions of the language $\mathcal{L}$ of arithmetic (terms, formulas, propositional functions, finite sequences of formulas, etc.) are an *effectively enumerable* set. This means that there is a mechanical procedure (an *algorithm*) which establishes a correspondence between linguistic expressions and natural numbers; given a linguistic expression we can compute a number which codes it, called its *Gödel number*; if A is a linguistic expression, we denote the Gödel number of A by $\ulcorner A \urcorner$.

Recall the following definitions:

- A set of numbers S is *decidable* if there exists an algorithm such that, given a number $n \in N$ decides whether or not n belongs to S.

- A function $f : N \to N$ is *effectively computable* if there exists an algorithm such that, for each argument n, computes the value $f(n)$ (an similarly for n-ary functions).

- To every subset S of N, we can associate the *characteristic function* $\chi_S : N \to \{0,1\}$ such that $f(n) = 1$ if $x \in S$ and $f(n) = 0$, otherwise.

Thus a set is decidable if and only if its characteristic function is effectively computable.

If Church's thesis is true, then in the above definitions *effective computability* is synonymous with *Turing-computability.* It is possible to prove the following theorem:

Representability Theorem. *For every Turing-computable function $f : N \to N$, there exists a propositional function $A(x, y)$ in the language $\mathcal{L}$ of arithmetic such that for all m, $n \in N$*

- if $f(m) = n$ then $A(\underline{m}, \underline{n})$ is derivable from Peano's axioms;

- if $f(m) \neq n$ then $\neg A(\underline{m}, \underline{n})$ is derivable from Peano's axioms.

A similar result holds also for n-ary Turing-computable functions. Intuitively, this means that the computation of the values of f can be performed as a formal computation within the deductive system of Peano. In this sense f is *arithmetically representable* by the propositional function $A(x, y)$.

It is actually convenient to expand the language $\mathcal{L}$ adding to it a name for each Turing-computable function. Thus, whenever we verify that f is a computable function, then we assume that there is a name "f" for it in the language of arithmetic $\mathcal{L}$. In this way we can reformualate the representability theorem in the following simplified form:

If $f : N \to N$ is a Turing-computable function, then for each pair m, $n \in N$

- if $f(m) = n$ then $f(\underline{m}) = \underline{n}$ is derivable from Peano's axioms;

- if $f(m) \neq n$ then $f(\underline{m}) = \underline{n}$ is derivable from Peano's axioms.

In particular, a decidable set S is representable by a propositional function $A(x)$ such that

- if $n \in S$ then $A(\underline{n})$ is derivable from Peano's axioms;
- if $n \notin S$ then $\neg A(\underline{n})$ is derivable from Peano's axioms.

(To see this, just replace $A(\underline{n})$ with equation $\chi_S(\underline{n}) = \underline{1}$, where χ_S is the Turing-computable characteristic function of S.)

1.2.1 Diagonalization

Diagonalization Lemma. *For every propositional function $A(x)$ there exists a number k such that $k = \ulcorner A(\underline{k}) \urcorner$.*

Proof. Let $f : N \to N$ be the function defined as follows. Given a number n,

$$\begin{aligned} f(n) &= \ulcorner B(\underline{n}) \urcorner, && \text{if } n = \ulcorner B(x) \urcorner, \text{ for some propositional function } B(x); \\ f(n) &= 0, && \text{otherwise.} \end{aligned}$$

Notice that f is effectively computable according to the following instructions: given n, decode it and find the corresponding expression; if this is a propositional function with one a free variable, then substitute the numeral $\underline{n}$ for the free variable x in $B(x)$ and compute the Gödel number of $B(\underline{n})$; otherwise, let $f(n) = 0$. Moreover, we can indeed produce a primitive recursive function f that does precisely this job. Therefore f is arithmetically expressible.

Now given a propositional function $A(x)$, letting $h = \ulcorner A(f(x)) \urcorner$, we have

$$f(h) = \ulcorner A(f(\underline{h})) \urcorner = \ulcorner A(\underline{f(h)}) \urcorner$$

(using the fact that f belongs to the language $\mathcal{L}$). This means that the code of $A(\underline{f(h)})$ is precisely $f(h)$ (just look at the definition of f). Hence $k = f(h)$ is the required number. The proof is finished.

Intuitively, the lemma expresses the existence of "self-referential" statements, in the following sense: if we interpret the numeral $\underline{k}$ as a name of the sentence $A(\underline{k})$ which it codifies, then this sentence says of itself (referring to itself through the name k) that it has the property $A(x)$.

1.3 Tarski's Theorem

Theorem. *There exists no propositional function in the language $\mathcal{L}$ of arithmetic that is satisfied exactly by those numbers which code sentences that are true in N.*

Proof. Suppose, for the sake of contradiction, that there exists a propositional function $V(x)$ in the language of arithmetic which is satisfied exactly by the numbers coding sentences that are true in N. By applying the diagonalization lemma to the sentence $\neg V(x)$, we find a number k such that $k = \ulcorner \neg V(\underline{k}) \urcorner$. There are two cases:

- If $\neg V(\underline{k})$ is true, then k is not the code of a statement true in N; but k is the code of the statement $\neg V(\underline{k})$, therefore $\neg V(\underline{k})$ is false, and this is a contradiction;

- If $\neg V(\underline{k})$ is false, then $V(\underline{k})$ is true, hence k is the code of a sentence true in N, and therefore $\neg V(\underline{k})$ is true in N, again a contradiction.

We have obtained a contradiction from assuming that such a $V(x)$ exists. The proof is finished.

In particular, the set of all true sentences in the language of arithmetic is not decidable: indeed, as we saw above, the decidable subsets of N are arithmetically expressible. Thus there can be no algorithm to decide for each arithmetical sentence, whether it is true or false.

Notice that the above proof can be regarded as a formal version of the lier paradox: the sentence $\neg V(\underline{k})$ says of itself that it is false. However, this is not a paradox, but simply a proof by contradiction.

Of course, the fact that the arithmetical truth-predicate is not expressible by a propositional formula of the language $\mathcal{L}$ of arithmetic does not prevent us from extending the language by introducing a new unary predicate symbol to be interpreted as a truth-predicate. Thus let us extend the language $\mathcal{L}$ to a language $\mathcal{L}' = \mathcal{L} \cup \{V(x)\}$. How shall we extend the standard model N to interpret the new symbol $V(x)$? There are two possibilities:

1. We may intepret $V(x)$ by the property of being the code of a sentence of $\mathcal{L}$ true in N;

2. we may interpret $V(x)$ by the property of being the code of a sentence of $\mathcal{L}'$ true in N.

If we adopt the second possibility, then we can actually reproduce the lier paradox within our formal system. Hence, this interpretation is ill-defined. However it is certainly possible to follow the first definition: indeed in this case $\neg V(x)$ is not a formula in the language $\mathcal{L}$, thus its code k does not satisfy the predicate V (since this is satisfied only by the codes of true sentences *in the language* $\mathcal{L}$). Hence in this case $\neg V(k)$ is simply *true* and no contradiction follows from this. Sure enough, we can extend $\mathcal{L}'$ with a new truth predicate

$V'(x)$ which will be true exactly of the formulas of $\mathcal{L}'$ which are true in N, but $V'(x)$ will belong to a new language $\mathcal{L}'' = \mathcal{L}' \cup V'(x)$, and so on.

1.4 Gödel Incompleteness Theorem

Gödel incompleteness theorem highlights an essential limitation of the axiomatic method: no mathematical theory, which is rich enough to incude arthmetic, can be axiomatized as a *consistent and complete* theory, i.e., a theory where for every sentence A either A or $\neg A$ follows from the axioms, but not both.

We shall consider Gödel's proof in the specal case where T is a decidable set of axioms in the language $\mathcal{L}$ of arithmetic, including Peano's axioms, and such that the axioms of T are true in the standard model. Notice that it is an essential property of an axiomatic system that one should be able to decide whether or not a sentence is an axiom: indeed to check that a proof is correct, one must be able to decide whether all its assumptions are axioms of the theory.

The proof of Gödel's theorem reproduces Tarski's argument, using the notion of formal provability in T instead of the notion of truth. Unlike the set of sentences true in N, the set of numbers which are codes of sentences *derivable in* T is arithmetically representable. Indeed we may enumerate all sequences s of expression of the language $\mathcal{L}$; since the set of axioms is decidable and so are the deduction rules, we can always decide whether or not a finite sequence s is a proof of a formula A. As a matter of fact, we can produce a primitive recursive binary function f such that

- $f(m,n) = 1$, if m is the code of a proof of a formula A such that $n = \ulcorner A \urcorner$,
- $f(m,n) = 0$, otherwise.

Therefore by the arithmetic representability theorem, there exists a propositional function $P(y,x)$ such that for all $m, n \in N$,

- $\vdash_T P(\underline{m}, \underline{n})$ (i.e., $P(\underline{m}, \underline{n})$ is formally derivable in the theory T), if $f(m,n) = 0$;
- $\vdash_T \neg P(\underline{m}, \underline{n})$, if $f(m,n) = 1$.

It follows that the expression $D(x) =_{df} \exists y.P(y,x)$ expresses the *provability predicate*:

(*) for every $n \in N$, $D(\underline{n})$ is true in N if and only if n is the code of a sentence provable in T.

Gödel's Incompleteness Theorem. *If T is a consistent theory whose set of axioms is decidable, true in the standard model and includes Peano's axioms, then there exists a sentence G which is true in the standard model N but such that neither G nor $\neg G$ is provable in T.*

Proof. By the diagonalization lemma, we can find a number k such that $k = \ulcorner \neg D(\underline{k}) \urcorner$. Let $G =_{df} \neg D(k)$. In the syntactic interpretation, G says that the sentence with Gödel number k (i.g., G itself) is not provable. Now if T is consistent, then G is certainly unprovable. Indeed,

suppose $\vdash_T G$:
then by $(*)$ $D(\underline{k})$ is true,
and therefore for some j, $f(j,k) = 1$ (the search for such a j terminates);
hence $\vdash_T P(\underline{j}, \underline{k})$ by arithmentic representability;
but from this we can derive $\vdash_T \exists y.P(y, \underline{k})$, hence $\vdash_T D(\underline{k})$, i.e.,
$\vdash_T \neg G(\underline{k})$ and so T is inconsistent.

Hence, if T is consistent, then G is unprovable, hence true. It follows that its negation $D(\underline{k})$ is false; but as we assumend that all axioms of T are true in the standard model and the rules of inference of Peano Arithmetic allow us to infer true formulas from true formulas, it follows that $D(\underline{k})$ is also unprovable in T. Thus we conclude that T is incomplete and that G is undecidable. The proof is finished.

Notice that in proving the unprovability of G we have also recognized its truth (after all, we proved it). This may seem paradoxical, but it is not. In fact what we have shown is that G is true *in the standard model* (assuming that the axioms of T are true in the standard model N). Indeed Gödel numbers are standard numbers and it is in the standard interpretation that the propositional function $D(x)$ represents deducibility from T. But being true in N does not imply being a logical consequence of T. It follows from the completeness theorem of first order logic that all sentences formally derivable (by first order logic) from T are those which are true in *all* models of T. Thus on the semantical level, the undecidability of G means that G is false in some non-standard model. Thus there exist non-standard models of Peano Arithmetic which are not elementary equivalent to the standard model N.

Of course, once we have acknowledged the truth of G we mak extend T to a theory $T' = T \cup \{G\}$. But by repeating the same argument, from the new set of axioms we will be able to produce a new statement G' which is true in the standard model and unprovable in T'. No extension of T can produce a system of axioms capable of determining all sentences true in N

as its logical consequences. The phenomenon of deductive incompleteness is therefore an essential limitation of the axiomatic method.