0.1 The incompleteness theorems

The incompleteness theorems are the most important results in the foundations of mathematics. These theorems are well known, but still many mathematicians do not fully realize their consequences. It is, perhaps, because they do not have much to do with the everyday work of a mathematician. But for those who study the foundations of mathematics, those who wonder what mathematical reality is and what mathematical truth is, this is the most important information. The incompleteness phenomenon ruins the naive hope that we could build a formal system for whole mathematics that would be complete and for which we could prove the consistency. Instead, we are facing a much more complex situation. In spite of the many decades that have passed since this breakthrough result appeared, we have not been able to find satisfactory answers to all the problems raised by the incompleteness theorems.

In this section I will be mainly concerned with the proofs of these theorems. They are a kind of results that can be easily explained on an informal level. But one has to be very careful when using imprecise descriptions of mathematical results and such descriptions must not be confused with real proofs. An informal argument used instead of proof may easily lead to false statements, especially in this case. Such wrong deductions, unfortunately, have been used to derive "consequences" of Gödel's theorems that have little to do with reality.

Let us recall the theorems.

Gödel's Incompleteness Theorems

1. *Given an explicit set of true arithmetical sentences, it is possible to construct a true arithmetical sentence that is not derivable from the set.*
2. *Given an explicit consistent theory T containing a sufficient amount of arithmetic axioms, the sentence expressing the consistency of T is not derivable from the theory.*

I am using the word *'explicit'* to denote sets of sentences that are algorithmically decidable and theories whose sets of axioms are algorithmically decidable.

Self-reference

One of the key ingredients of Gödel's proof is what is often called *self-reference.* This is a situation in which some subject expresses facts about itself. More generally, we may include also situations in which a program processes its own code and similar ones. Why self-reference is important is clear—the proof is based on the liar paradox and for that we need a kind of circular statement. Let us look at the following statement.

This sentence has 24 letters.

The sentence refers to itself by the word *'this'*. In natural languages we do not have problems stating such sentences, but in formal systems the language is restricted and it is not clear if one can make such statements. What precisely does it mean *'this'* in the sentence above? We know that it refers to the sentence itself, but we deduce it from the context (there is nothing else to which the word might point). Let us take another sentence.

> *The following sentence has 24 letters "It's a beautiful day today.".*

(The sentence is false, but this is of no concern now.) Here we point to a sentence by words *'the following'*. This way of referring to a sentence is much less ambiguous. It is customary to refer to a sentence by putting it between quotation marks, thus the sentence is determined uniquely. Furthermore, the sentence to which we refer is given more explicitly. In the first case, we had to finish reading and then we had to go back to read the sentence that is referred to. This is not the usual way one reads the text. Returning back to the beginning of the sentence is not the main problem. What matters more is that the second time we read it we do not read the sentence in order to get a message but we take it as plain text, as a sequence of symbols. While in the first case we may doubt whether we should allow such constructions in formal systems, because it leads to paradoxical sentences such as:

> *This sentence is false.*

the second kind of referring to sentences seems quite innocent. Often the first thing that you learn when studying a programming language is how to make a computer to print a sentence on the screen, for example:

```
println(''Hello world!'')
```

This is another example of a direct reference, we are not greeting the computer, we are asking the computer to greet. In fact, even if we wanted we could not forbid such a direct reference in programming languages. A sentence is merely a sequence of letters; we cannot prohibit talking about sequences of concrete symbols. Input data for a computer are such sequences, numbers represented in decadic or binary forms are such sequences, etc.

So let us assume that direct reference is possible in a formal language that we are using in logic. This can be easily checked for concrete versions of logical calculi, but we do not have to do it, as we know that a system without this property would be very limited. Contrary to what the above examples may suggest *direct reference* alone enables us to write down *self-referential* sentences. Of course, we need a little trick. To discover this trick let us try to do it directly. Our first attempt would look like this:

> *The following sentence has 32 letters "The following sentence has 32 letters.".*

This is true, but the sentence does not refer to itself. It refers to a sentence that is only a part of it. Doing it in such a straightforward manner we will never succeed, for the whole sentence must always be as long as the sentence that we refer to. The sentence above is twice as long as the sentence that it refers to.

But there is something even more conspicuous in that sentence: it consists of two copies of the same sentence, some quotation marks and periods. Perhaps, we can make use of this property? Yes we can! Look at this:

> *The following sentence written once with no modification and then once again between the quotation marks has 100 letters "The following sentence written once with no modification and then once again between the quotation marks has 100 letters"*

This is a sentence that speaks about itself. (To make it simpler I left out the periods, thus it is not quite grammatically correct, but this is only a minor point.) It is clear that this trick works in general, we can say whatever we want about the sentence. It is worthwhile to repeat it with the liar paradox sentence.

> *The following sentence written once with no modification and then once again between the quotation marks is false "The following sentence written once with no modification and then once again between the quotation marks is false"*

Self-reference is a very interesting phenomenon, but what we need for the proof of Gödel's Theorem is merely a simple technical trick based on saying that the sentence should be written twice. So do not confuse the incompleteness, which is the impossibility of proving all true sentences, with the impossibility of referring to itself. The latter is false, sentences *can* refer to themselves! So can also axiomatic theories express things about themselves (but for this we even do not need any special tricks). The reason why people often confuse these two things is the following. In order to resolve the liar paradox one has two possibilities, either to say that it is caused by self-reference or by the impossibility of defining truth for every sentence. We have seen that it is impossible to forbid self-reference without severely restricting the language. Hence the problem is in the definition of truth. For some people this seems less acceptable, therefore they mistakenly think that self-reference is the problem.

To conclude this section, here is a simple exercise in programming in which you can check the possibility of self-reference in a computer language.

Exercise. *Write a program (in your favorite language) that prints its own code. (The program should run on any machine, so you cannot cheat by telling the program the place where the code is stored. Check the Internet for the shortest known such program!)*

Arithmetization of syntax.

When trying to learn Gödel's theorems you may have come across such a rather strange title. Maybe, that was the reason why you gave up reading on. But there is nothing deep behind it, and my aim in this section will be to show you that this is not such an essential part of the argument as often believed to be.

Let me start with *formalization of syntax*. By this we mean giving a precise mathematical definition of the concept of a proof and proving the basic properties of this concept. From Chapter 2 we know that proofs can be rigorously defined. Furthermore, we know that essentially all mathematics can be formalized in Zermelo-Fraenkel Set Theory. Thus we have a formal definition of axiomatic systems, and we have one axiomatic system in which we can formalize axiomatic systems. In particular in this system we are able to reason about the system itself. This suffices for the proof of the First Incompleteness Theorem. In particular, it enables us to prove the First Incompleteness Theorem for Zermelo-Fraenkel Set Theory. Notice that we have only used that Zermelo-Fraenkel Set Theory is strong enough to prove basic properties about the syntax of logic. Hence, if we take a stronger theory, we will be able to prove its incompleteness again. But if we take a weaker theory, then it is trivially incomplete, since it does not prove all the axioms of Zermelo-Fraenkel Set Theory.

Thus as long as we are only interested in proving the incompleteness of set theories compatible with Zermelo-Fraenkel Set Theory, we can simply refer to the fact that logic can be formalized. When we want to prove the second incompleteness theorem, we have to be more careful. In order to prove that the consistency of a theory T is not provable in T, we have to formalize syntax in T. It does not suffice to do it in an extension of T. So the question of what is the minimal strength of axioms which suffice for formalization is relevant. One can show that we do not need such a strong theory as Zermelo-Fraenkel Set Theory, it suffices to use Finite Set Theory (which is Zermelo-Fraenkel Set Theory less the Axiom of Infinity). The reason is that syntactical entities are finite, hence we only need finite sets. Hence the Second Incompleteness Theorem can be proven for all extensions of Finite Set Theory.

Syntax can be formalized not only in set theories, but also in arithmetical theories. When we formalize the concept of a proof in arithmetic, we call it *arithmetization of syntax.* We do not get such a formalization automatically as in set theory. In arithmetic we do not have the concept of a sequence of symbols, we have to simulate it by a number-theoretical concept.

Gödel first proved the First Incompleteness Theorem for the Theory of Types, the system introduced by Russell and Whitehead in *Principia Mathematica*. In this theory a logical calculus can be represented directly, which means that the definitions of formulas and proofs can be literary used—in the same way as in Zermelo-Fraenkel Set Theory. Gödel used such a natural formalization in his proof—numbers to represent symbols, sequences of numbers to represent formulas and sequences of sequences of numbers to represent proofs. (The reason for choosing the Theory of Types instead of Zermelo-Fraenkel Set Theory was simply the fact that the latter was not so well established at that time as it is now.) So Gödel did not use arithmetization of syntax in his first proof. The resulting independent sentence was a sentence of set theory. After presenting his proof in Königsberg in 1930, von Neumann asked him whether he could construct an independent sentence that was purely arithmetical. It was a challenge, since such sentences would concern number theory, the heart of mathematics, rather than set theory, an invention of logicians. Gödel succeeded in finding such sentences soon after that. To this end, he had to represent all syntactical objects by numbers, not only symbols, in other words, he had to arithmetize syntax.

In retrospect this does not seem like a very difficult task. We are now used to the fact that all finite structures can be encoded by strings of bits. Strings of bits can readily be interpreted as digits of numbers. But that was more than a decade before first digital computers appeared.

In his fundamental paper Gödel stated the theorems for the Theory of Types of Russell and Whitehead, but in fact he used the Theory of Types only as an ambient logic for arithmetic. The proof works without modification for the theory that we call Peano Arithmetic.

Let me briefly sketch the main ideas of arithmetization of syntax. The key observation is that syntactical objects can be viewed as texts, which are sequences of symbols. We start with identifying the finitely many symbols of logic with some numbers. Say, if we have 26 symbols, we can identify them with numbers $1, 2, \dots, 26$. Then we need to code sequences of symbols, which means to code sequences of numbers. The most popular way is to use exponents in prime factorizations of numbers. In this encoding the number 1500 represents the sequence $(2, 1, 3)$, since $1500 = 2^2 \cdot 3^1 \cdot 5^3$. Since Peano Arithmetic does not have a symbol for the operation of exponentiation, first we have to show that exponentiation can be defined. This is a little tricky and Gödel invented an ingenious way of defining sequences without exponentiation (see page 20). Anyway it is only a technical detail, since if we include exponentiation as a basic operation, we can avoid this complication.

Formalization does not mean only to assign certain sets or numbers to syntactical entities. We also need to prove that the representations behave as they should. Furthermore, we need to formalize syntactical concepts by formulas of low quantifier complexity. This is the reason why the incompleteness theorems require that the set of axioms of the theory is decidable.

To summarize the preceding paragraphs, we need formalization of syntax for the incompleteness theorems, but it does not have to be arithmetization. If we want to prove the theorems for set theories, we can use standard formalizations, if we want to prove them for theories of arithmetic, we have to use formalization based on numbers, called arithmetization, and if we wanted to prove them for some other theory, we would need a formalization based on the concepts of that particular theory.

This brings us to the natural question: which are the theories to which incompleteness applies? First of all, we have to distinguish two kinds of theories. The first kind are theories that define a class of structures and the aim is to define a large variety of structures of certain type. For example, we define groups by a set of equations, but we do not want to decide properties, such as the commutativity of the group operation. We want to study commutative groups as well as non-commutative ones. Thus such a system of axioms is designed to be incomplete.

The incompleteness of the type that appears in Gödel's Theorem is different. It applies to theories that attempt to describe a single structure, such as the universe of sets, the algebraic structure of natural numbers or real numbers.

There we would like to get a complete theory, but sometimes it is impossible. It is this second kind of theories for which we use Gödel's theorems, but not all such theories are incomplete. There are important mathematical structures that can be completely axiomatized.[1]

It is difficult to state exactly for which theories T one can use Gödel's argument, since the proof can be generalized in various ways. The following two properties are essential in any case:

1. *theory T is consistent, and*

2. *the axioms of theory T are explicitly presented.*

In the original proof Gödel required the condition that T be a true theory, which is stronger than consistency. Recall that property 2. means that there is an algorithm for deciding if a sentence is an axiom of T. In particular, this is the case if the theory has a finite number of axioms. The axioms of theories used in the foundations of mathematics are always presented quite explicitly, so usually this property is obvious.

We need a third condition for both theorems, but the conditions are different for the two incompleteness theorems. For the First Theorem we need:

3. *T is compatible with a theory in which it is possible to formalize syntax.*

This means that there exists a theory S such that the primitive concepts of S and T are the same and the union of the two theories is consistent.

For the Second Theorem we simply need:

$3'$. *it is possible to formalize syntax in T.*

Conditions 3. and $3'$. are stated informally, since the I have not explicated what formalization of syntax means. This can be made precise by saying that there is a formalization of the provability predicate that satisfies certain *derivability conditions* (see page 23). Condition 3. is not very useful for applications, it is much better to give some minimal requirements about what should be provable in theory T. One can show, in particular, that it suffices that T proves some very basic properties of arithmetical operations, or in case of set theories, some very simple sentences concerning sets.

The proof of the First Incompleteness Theorem

Now that we have all the ingredients for the proofs of the incompleteness theorems, we can start with a proof of the First Incompleteness Theorem. In order to make the proof a little simpler, I shall assume that T is a true theory. This means that the axioms of T, whence all theorems of T, are true statements.

We need to construct a sentence that is not provable in T. It will a selfrefential sentence, which I will call γ_T. Informally, the sentence expresses the following:

[1] An example is $(\mathbb{R}; +, \times, <)$ the structure of real numbers with operations $+$ and $\times$ and the relation $<$. The true sentences in this restricted language can be recursively axiomatized.

This sentence does not have a proof in T.

Stated more precisely, γ_T is a sentence such that the following is provable in T:

γ_T is true if and only if γ_T does not have a proof in T.

In the preceding pages I have explained why it is possible to write down such a sentence in a formal logical language.

First I will prove that γ_T is true. This is proved by arguing by contradiction as follows. Suppose that γ_T is provable in T. Then γ_T is false, since γ_T asserts that γ_T is not provable. On the other hand T can only prove true sentences, hence assuming that γ_T is provable, γ_T must be true. So, assuming that T proves γ_T, we have proved that γ_T is false and true at the same time, which is a contradiction. Thus γ_T is not provable.

Once we know that γ_T is not provable we are done, because we have

1. γ_T is unprovable, and

2. γ_T is true, because it expresses the statement above.

Thus we have proved the First Incompleteness Theorem. The table below should help reader fully understand what is going on in the proof.

	γ_T is true	$\neg\gamma_T$ is true
γ_T is provable in T	**impossible:** γ_T asserts that γ_T is unprovable in T	**impossible:** only true sentences are provable in T
γ_T is unprovable in T	the only **possible** case	**impossible:** $\neg\gamma_T$ asserts that γ_T is provable in T

The proof resembles very much the liar paradox, more precisely, the paradoxical sentence *'This sentence is not true.'* The difference is, of course, in using the predicate *'provable in T'* instead of *'true'*. The concept of a proof in a formal theory is a precise mathematical concept. Moreover, it is a combinatorial concept, a concept that does not require infinite sets. On the other hand the concept of the truth is not a syntactical concept and thus cannot be defined in combinatorial terms. We will see below that actually this paradoxical sentence proves limitation on how truth can be defined.

The proof of the Second Incompleteness Theorem

The second incompleteness theorem asserts that, assuming that T is a sufficiently strong theory, T cannot prove the sentence that asserts that T is consistent. Hence the second incompleteness theorem extends the first one by giving an explicit statement that is unprovable. Furthermore, the unprovable sentence is a very important one, it concerns the problem of consistency. Note that the proof of the first incompleteness theorem does give quite a concrete independent sentence, but is a rather strange sentence that does not tell us anything interesting.

Let us denote by Con_T the statement that T is consistent. We want to prove that T does not prove this sentence. The idea of the proof is to show that Con_T implies the Gödel sentence γ_T. Then Con_T cannot be provable in T, because otherwise also γ_T would be provable and we know that the latter is false. To prove that implication, we do not need any technicalities. This is done simply by analyzing the proof of the First Incompleteness Theorem, but we need a different proof of that theorem. We need a proof of the First Incompleteness Theorem that uses only the consistency of T instead of T being a true theory.

The proof of the First Incompleteness Theorem based on the consistency of T uses the same sentence γ_T. By way of contradiction, suppose that γ_T is provable in T. Let d be a proof of γ_T in T. Depending on how we formalize syntax, d is a concrete finite set, number, or a finite string. Checking that d is a proof can be done by a simple procedure. Assuming T is a sufficiently strong theory, we can formalize the steps of the procedure applied to d, and thus we show that T can verify that d is a proof of γ_T. Whence it is provable in T that

> γ_T *is provable in* T.

But this statement is equivalent to $\neg\gamma_T$ in T. Hence $\neg\gamma_T$ is provable in T. Putting it together with our assumption that T proves γ_T, we get that T is inconsistent.

This contradiction finishes the proof of the First Incompleteness Theorem (with the weaker assumption that T is consistent).

To prove the Second Incompleteness Theorem we first observe that the above proof gives us a proof of the implication:

> *If* T *is consistent, then* γ_T *is not provable in* T.

Now recall that γ_T says that *"γ_T is not provable in T"*. Hence the above sentence can be simplified to

> *If* T *is consistent, then* γ_T.

Thus, indeed, the sentence Con_T implies the sentence γ_T. Hence Con_T is also not provable in T. This finishes the proof of the Second Incompleteness Theorem.

Although we do not need it for the proof, it is worthwhile noting that γ_T is *equivalent* to Con_T. This means that each formula implies the other. We have already shown that Con_T implies γ_T. The converse implication is even simpler. The sentence γ_T asserts that something (namely γ_T) is not provable in T. But if some sentence is not provable, the theory must be consistent, for in an inconsistent theory everything is provable. Hence γ_T entails Con_T.

The proofs above show that γ_T and Con_T are not provable in T, provided that T is consistent. However, we cannot conclude that T is incomplete, for it can happen that $\neg\gamma_T$ and $\neg Con_T$ are provable. Indeed, take a theory S to which Gödel's Theorems apply and extend it by the axiom $\neg Con_S$. This theory is consistent, because its consistency is equivalent to the statement that S does not prove Con_S (the negation of the added axiom), which is the content of is the Second Incompleteness Theorem. Let T be the theory S extended with $\neg Con_S$. Since the inconsistency of S entails the inconsistency of T, T proves its inconsistency $\neg Con_T$. Now Con_T is equivalent to γ_T, so $\neg\gamma_T$ is also provable in T.

Thus Gödel's Theorems did not exclude the possibility that one could find an extension, say of Peano Arithmetic, which would be a consistent complete theory (for every sentence ϕ, either ϕ or $\neg\phi$ would be provable in the theory). The theorems only imply that such an extension could not be a true theory (in particular, the theory would have to prove its inconsistency). But one can construct just a little bit more sophisticated self-refential sentence ρ_T such that neither ρ_T nor $\neg\rho_T$ are provable in any consistent T (see *Rosser's formula*, page 18).

The undefinability of truth

Gödel showed the impossibility to realize Hilbert's program by his second incompleteness theorem. Interestingly enough, he discovered it while working in the direction that Hilbert proposed. Gödel wanted to show that assuming Peano Arithmetic, a first order theory of arithmetic, is consistent, then also a second order theory of arithmetic is consistent. The latter theory is not presented using second order logic, it is an axiom system in first order logic that extends Peano Arithmetic by introducing new objects that are sets of natural numbers (including infinite ones). In this system every formula that speaks on numbers defines a set. Thus Gödel decided to represent sets of numbers first by formulas and then, by encoding formulas by numbers, to represent sets by numbers. He needed a number-theoretical predicate that would represent the relation *'number n is an element of set X'*. As X was represented by a formula with a free parameter for numbers, say $\phi(x)$, he needed to find a formula that would express the binary relation *'$\phi(n)$ is true'* for every formula ϕ and every number n. But then he realized that having such a formula for truth, he would be able to do the liar paradox in Peano Arithmetic and thus get a contradiction in Peano Arithmetic. So, assuming the theory is consistent, one cannot find such a formula. Then he probably continued as follows. He realized that if he used the predicate of being *provable in Peano Arithmetic,* instead of the predicate of being *true,* he would be able to construct a corresponding formula. Thus, knowing that there was no formula that defined truth and knowing that there was one that defined provability, he concluded that the two things must be different, truth cannot be equal to provability. There must be true unprovable sentences.

Analyzing this argument one can easily see that in order to prove incompleteness we do not have to mention the predicate of being true and we can apply the self-reference directly to provability. Thus in Gödel's paper the fact about the undefinability of truth does not appear. Later Alfred Tarski, who was more interested in models than in proofs, rediscovered this result.

The theorem about the undefinability of truth concerns the same theories as Gödel's First Incompleteness Theorem, since it only needs the construction of self-referential sentences. But, unlike Gödel's theorems, it can also be stated for models, namely those that satisfy the axioms of a sufficiently strong theory. I will concentrate on the version for models. Here is the explicit statement (except that I am again using the rather vague assumption about being sufficiently strong).

The Gödel-Tarski Theorem *Let T be theory in a language L, sufficiently strong to be able to speak about formulas. Let M be a model T. Then there is no formula $\phi(x)$ in language L with a free variable x, such that for every formula ψ in language L and a term t representing ψ, ψ is true in M if and only if $\phi(t)$ is true in M.*

Unfortunately, we have to state it in such technical terms; it is a rather subtle subject. There are a couple of new things here that we have disregarded in the presentation of the incompleteness theorems, because they were not so important there. It is the mentioning of a language and terms. The language plays an important role here, as it is possible to extend the language to get a definition of truth. So it is important that the potential definition of truth $\phi(x)$ is from the same language as the formulas ψ that it is supposed to cover. On the other hand the role of representing formulas by terms is minor. It is only the technical fact that in the usual logical calculi we cannot refer to a formula directly by plugging it into another formula (given formulas ϕ and ψ, $\phi(\psi)$ is not a legal formula). It is the same thing as using quotation marks when referring to a sentence as plain text. As we are concerned about using only language L we have to be careful about how we refer to a formula. That's why I have written it in this more complicated way. The meaning of the theorem is, however, simple:

> *for formulas in language L, it is not possible to define truth in model M by a formula in the same language L and interpreting the formula in the same model M.*

The only technical part of the proof is writing the self-referential formula *'this formula is false'*. The difference between this proof and the proof of the first incompleteness theorem is that now we do not have the formula for truth, it is what we want to disprove. We argue by contradiction and assume that some $\phi(x)$ is such a formula. Then we construct the self-referential formula for it and we get a contradiction.

The theorem on the undefinability of truth is not in contradiction with what I said in Chapter 2. There I said that truth was definable, and we used that to define tautologies, and eventually to show that a logical calculus for first order logic is complete, etc. Thus there are certain situations when truth is definable and when it is not. As you surely noticed, I was very careful when stating the theorem on the undefinability. Given a structure M for which we need a truth definition, the two situations are distinguished by the point of discourse (the position from which we talk about the structure). If we are inside of the structure it is not possible. If we are outside it is possible, provided that the outside environment is sufficiently complex. Let me state it explicitly.

- It is not possible to define truth about M inside of M.
- If we argue about the structure in a sufficiently strong set theory where M is a set (equipped with relations and function symbols), we can define truth about M.

We can think of this situation also as follows. Suppose we have a model of set theory M. In this model we cannot define truth about itself, but for a structure N that is a set in M we can do it. Though we cannot define truth for all M we can define it for a part of it.

The proof of the undefinability of truth also explains in which context the definition is possible and where it is not. The proof requires self-reference, in order to get (a version of) the liar paradox. If we talk about a structure from outside, there is no self-reference.

The theorem does not play such an important role as the incompleteness theorem. Rather than having deep consequences, it provides explanations. Firstly, it explains semantical paradoxes such as the liar paradox, (explanation being that truth is not definable) and Berry's paradox (explanation being that 'definable' is not definable). Secondly, it helps us understand why a theory cannot prove its own consistency. The most common way of proving the consistency of a theory is to provide a model of it. To use the model to prove that the theory is consistent we need a definition of truth for the model. In set theory, if the model is a set, there is no problem. We can also define class models, but for such models truth is not definable, because "classes are too big". In particular, the universe of all sets can be viewed as a class model. If we could define truth for it, we could use this class model to prove the consistency of the theory in itself. So the explanation that the theorem provides is roughly the following. The universe in which we argue is too big to be described by itself, hence there is no way to prove that it is consistent.

This may suggest that the theorem about the undefinability of truth says that, after all, a certain type of self-reference is impossible, namely, in a language L we cannot define the truth about L. It is disputable whether this should be called self-reference, but words are not important. The difference between the self-reference used in the proofs of Gödel's theorems and the undefinability of truth is that the former concerns syntax whereas the latter concerns semantics. In a given language we may be able to define its syntax, but not its semantics.

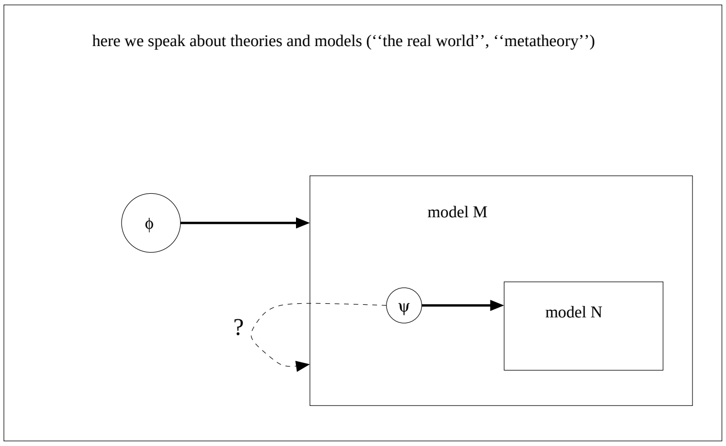

Figure 1: A picture showing the hierarchy of systems in which we speak about concepts of logic. The largest box is the highest level. This is the level on which we communicate; here we use a natural language, such as English, and we assume the most common mathematical principles, such as mathematical induction (however, it is possible to use a formal system, for example, a programing language, also on this level). At this level we define when a sentence ϕ is true in a model M. If M has sufficiently rich structure, we can define logical concepts inside of M; in particular, we can define when a sentence ψ is true in a model N which is defined in M. But in M we do not have a general definition of the truth of sentences in M itself.

Some peculiarities of incompleteness

People are often baffled by the fact that for an arithmetical formula $\varphi(x)$ it can happen that the sentence

$$\textit{for every } x,\ \varphi(x)$$

is unprovable, but all sentences

$$\varphi(0), \varphi(1), \varphi(2), \varphi(3) \dots$$

are provable. I am speaking, of course, about provability in a fixed theory T. When writing $\varphi(0), \varphi(1), \varphi(2), \varphi(3) \dots$ I assume that either T contains explicitly all numerals $0, 1, 2, \dots$, or we have some representation of them by suitable terms in T. If, for instance, T contains constants for 0 and 1 and the operation of addition, then we can use $0, 1, 1+1, 1+1+1, \dots$, thus we get

$$\varphi(0), \varphi(1), \varphi(1+1), \varphi(1+1+1) \dots$$

The first thing one should realize is that the first expression is a single sentence Φ, whereas the second one is an infinite set of sentences. Clearly, using basic logical rules we can derive every sentence of the set from Φ, but there is no rule in logic that allows us to do the opposite. Since all proofs in logic are finite, any proof can use only a finite number of premises. Hence, if we had a proof of the sentence Φ from the infinite set $\{\varphi(0), \varphi(1), \varphi(2), \dots\}$, it would use only a finite number of sentences of the form $\varphi(n)$, for n a numeral. But, clearly, a statement about all numbers does not logically follow from a finite number of special cases.

Suppose Φ expresses the fact that an algebraic equation does not have a solution in the domain of natural numbers. We know that it may be very difficult to prove such a sentence. For example, to prove Fermat's Last Theorem for the exponent 3,

$$(x+1)^3 + (y+1)^3 \neq (z+1)^3,$$

is a nontrivial task. However, to prove that a given concrete number is not a solution is trivial, everybody can do it. To this end you simply need to evaluate the terms of the equation and compare them. Such a computation constitutes a proof, in fact, it is the most basic kind of proof. To be quite precise, mathematicians accept computations as proofs, or parts of proofs, but computations are not explicitly included in the logical calculus. In order to be able to use computations as formal proofs, we have to assume as axioms some basic properties of the operations. In the case of arithmetic computations with integers we need a few simple axioms that are present in any reasonable axiomatization of the arithmetic of natural numbers. In particular the axioms of Peano Arithmetic, even without the axioms of induction, suffice for this purpose.

It is not difficult to find a sentence φ with these properties. As a matter of fact, every unprovable true sentence which is a universal finite formula provides us with such an example. Let us consider Gödel's formula Con_T which expresses that a theory T is consistent. It is a statement of the form

> *Every string of symbols is not a proof of contradiction from the axioms of T.*

Suppose T is consistent, then by Gödel's Theorem this sentence is unprovable in T, but for every given string of symbols we can verify that it is not a proof. This verification can again be written as a formal proof in T (provided that T is a sufficiently strong theory to prove basic properties of strings of symbols).

Flexible formulas and unpredictable programs

In the literature there are a variety of results inspired by Gödel's incompleteness theorem. One of these which strikes me most is the *flexible formula* of Andrzei Mostowski. Let T be a sufficiently strong theory that contains arithmetic. Then it is possible to construct a formula $\mu(x)$ with one free variable x with the following property. Not only is $\mu(n)$ independent from T for every numeral n, but also these sentences are *mutually independent*. This means that the axioms of T together with sentences of the form $\mu(m)$ or $\neg\mu(m)$ for $m \neq n$ are still unable to decide if $\mu(n)$ holds. Formally it is stated by saying that if we choose arbitrarily $\mu(n)$ or $\neg\mu(n)$, but not both, for every n, then this set of sentences together with T is consistent for every such choice. For example

$$T \cup \{\mu(0), \mu(1), \neg\mu(2), \neg\mu(3), \mu(4), \neg\mu(5), \mu(6), \neg\mu(7), \mu(8), \ldots\}$$

is a consistent theory.

The unpredictable program is an example of the same flavor. We already know that it is algorithmically undecidable whether a given program stops. If we are not able to decide if the program stops, can we at least prove something about what the program will do? For instance, can we at least prove that the program is not a virus that will damage our computer? The answer is again no. But again, as in all independence results, the unpredictability depends on the theory T in which we argue about the program.

Here is a program which, from the point of view of T, can print any string. More precisely, we cannot prove in T that the program will not print y, for any given string y.

Program $UNPR_T$**.**

1. `systematically search all proofs of` T`;`
2. `if you find a` T `proof of the sentence` *" $UNPR_T$ does not print y",* `for some string of symbols` y`, then print` y`.`

This is not an explicit definition of a program, because it refers to itself. But we already know how to replace self-reference by direct reference, so the above program can be written explicitly in every reasonable programing language.

Let us show that assuming that T proves only true sentences, it is consistent with T that $UNPR_T$ prints an arbitrary string y. Arguing by contradiction, let y be an arbitrary string and suppose that T does prove that $UNPR_T$ never prints y. Let d be this proof. It may happen that there are actually several proofs of this statement, moreover there may exist such proofs for other strings too. Thus we consider the order in which $UNPR_T$ searches proofs and take the first such proof d_0. Let y_0 be the string to which this proof refers to. Then, by definition, $UNPR_T$ must print y_0. But this is impossible, because T proves only true sentences. We got a contradiction, hence the assumption that T proves that $UNPR_T$ never prints the string y is false.

Looking at $UNPR_T$ from a higher perspective we can show that it will never stop, so it will not print any string. But notice that this requires the assumption that T is consistent. Indeed, if T is inconsistent, then T will stop and print some string, because if T is inconsistent, there are proofs of all sentences in T and it is only the matter of which string y will occur in such a proof first.

You may counter by saying that we can prove in T that $UNPR_T$ is quite innocent—it will either run for ever, or print a string and stop. So look at the following modification.

Program $UNPR_T^+$**.**

1. `run` $UNPR_T$`;`
2. `if` $UNPR_T$ `prints a string` y `which is a code of a program, then run` y`.`

Now you cannot prove, using only T, that this program will not destroy your computer.

Doing things formally may obscure the main ideas, but once the essence of the proof is clear, seeing more precise arguments will surely help the reader. The only technical part in the proofs of incompleteness theorems is constructing self-referential sentences, so I will concentrate on it.

As I said, writing correct formulas which express facts about formulas requires representation of formulas by terms. Suppose we have a theory T that contains arithmetic of natural numbers and we want to speak about a concrete number, such as 3 or 2001. It is possible that T contains decimal numerals as constants and there is no problem, but it does not have to be the case. In such a case we have to describe the number somehow. When the theory contains constants and function symbols, we may use terms. For example, using the constant 1 and the binary function symbol $+$ we write 3 as $1+1+1$. Some theories do not have constants nor function symbols. Then we still can describe a concrete number (0 is the least number, 1 is the next after the least, etc.). Also in programming languages some data can be written down directly (surely numbers) and some have to be described. When talking about syntactical objects we have to be more careful. We cannot assume that each syntactical object can be used also as the name of itself. This is not possible even in natural written language. Consider the sentences:

I am talking about you.

and

I am talking about "you".

We have to use quotation marks to distinguish *you* as a person from *you* as a word. To make this distinction in formal languages people use various notations. Here I will use $\ulcorner\ldots\urcorner$ as such quotation marks. Thus $\ulcorner\phi\urcorner$ is a term that represents the formula. In case of a theory that only speaks about numbers, first we represent formulas as numbers, and then $\ulcorner\phi\urcorner$ will be a term representing the number assigned to the formula ϕ. Should it sound confusing, think of $\ulcorner\ldots\urcorner$ simply as quotation marks.

The next thing is to represent functions. For instance, you want to talk about squaring numbers. If you have $\times$ in T, then you can use the term $x \times x$. Having only a small number of function symbols and constants we cannot hope to represent all functions. There will be a lot of functions that we can describe by formulas and not by terms. For the exposition of the proof it is, however, very convenient to assume that we can represent a certain function by a term. Namely, we need the *doubling* function

$$\sigma(x) \mapsto \sigma(\ulcorner\sigma\urcorner). \tag{1}$$

This is the function that given a formula $\sigma(x)$ with a free variable, substitutes the term representing $\sigma(x)$ for the variable x. I will denote this function by d and assume that it can be defined by a term in T (and I will use the same symbol for the term). Function d is the function *'write the following sentence twice... '* from the verbal exposition of self-reference in the previous section. Formally we need the following to be provable in T for every formula $\sigma(x)$,

$$d(\ulcorner\sigma(x)\urcorner) = \ulcorner\sigma(\ulcorner\sigma(x)\urcorner)\urcorner. \tag{2}$$

This is a formalization of the fact that d represents the function defined by (1). Now let $\phi(x)$ be an arbitrary formula. We shall prove an important lemma which is a formal statement of self-reference. The lemma is called the **Diagonal Lemma** or the **Fixed Point Theorem**.

Lemma 0.1 *Given a formula $\phi(x)$, it is possible to construct a formula ψ such that the following equivalence is provable in T*

$$\psi \equiv \phi(\ulcorner\psi\urcorner).$$

The meaning of ψ is the self-referential statement

This formula satisfies condition ϕ.

We already know how to do it:

The following formula written twice ... satisfies condition ϕ: "The following formula written twice ... satisfies condition ϕ".

(The dots indicate that I have not written it completely. One has to add that the second occurrence is after a colon and in the scope of quotation marks and finally there is a period.)

To do it formally, we use the doubling function d. Thus ψ is

$$\phi(d(\ulcorner\phi(d(x))\urcorner)). \tag{3}$$

It is worth checking that the formula expresses the same as the sentence above: $\phi(\ldots)$ stands for *satisfies condition* ϕ, $d(\ldots)$ for *written twice* and $\ulcorner\phi(d(x))\urcorner$ is the direct reference *"The following formula written twice ... satisfies condition* ϕ*"*.

Now we shall show that the sentence does what it should. Notice that ψ has the form $\phi(t)$, where t is the term $d(\ulcorner\phi(d(x))\urcorner)$. We need to prove in T that $t = \ulcorner\psi\urcorner$. Indeed, by (2) and (3),

$$d(\ulcorner\phi(d(x))\urcorner) = \ulcorner\phi(d(\ulcorner\phi(d(x))\urcorner))\urcorner = \ulcorner\psi\urcorner.$$

Hence ψ is equivalent to $\phi(\ulcorner\psi\urcorner)$.

Note that this proof is almost identical with the proof of the Fixed Point Theorem of λ-calculus, see page **??**. This is not surprising, especially for those who solved the problem on page 3. That exercise asks for a fixed point of the function computed by the compiler for the programming language—giving the compiler the program it simply prints the program.

For the proof of the incompleteness theorem we take a special formula for $\phi(x)$, the formula saying that x has no proof in T. Let us denote this formula $\neg\pi_T$. Then there exists a formula γ_T such that in T

$$\gamma_T \equiv \neg\pi_T(\ulcorner\gamma_T\urcorner).$$

To finish the proof we need one more fact: *if a sentence* ϕ *is provable in* T*, then* $\pi_T(\ulcorner\phi\urcorner)$ *is also provable in* T. This is a special case of the following general principle, called Σ-*completeness*:[2]

If ψ *is a true* Σ_1 *sentence, then* ψ *is provable in* T.

So if a true sentence is sufficiently simple, then it is provable. Here is the idea of why it is true. Consider the following very simple true sentence.

$$(1+1)\cdot(1+1) = 1 + (1 + (1+1)).$$

This equality can be easily derived using the distributive and associative laws. We surely have such laws in any reasonable arithmetical theory. Now consider the following Σ_1 sentence (a sentence asserting that 4 is even).

$$\exists x\; x\cdot(1+1) = 1 + (1 + (1+1)).$$

An important observation is that to derive this sentence from the one above we only used logic. So the same theory that proves the first sentence, proves the second one. In Σ_1 sentences we may also use bounded universal quantifiers, so we may have a sentence of the form

$$(\forall x \leq 1+1)\alpha(x).$$

Proving such a sentence reduces, using only some basic axioms of arithmetic, to proving three sentences

$$\alpha(0), \alpha(1), \alpha(1+1).$$

In such a way we can gradually reduce proving a Σ_1 sentence to proving simple equations and inequalities of the type above with no free variables. The proofs of such equations and inequalities are essentially numerical evaluations of terms.

[2] It would be more appropriate to call it Σ_1 completeness, but I will stick to the traditional notation.

Now we need to check that the concept of provability in T is Σ_1. This is the part of the proof that uses the fact that the set of axioms of T is decidable (which I called *'an explicit set of axioms'*). This condition on T is, in fact, stronger than what is needed for the proof; it suffices only to assume that the axioms of T form a recursively enumerable set (a Σ_1 set). However, axiom systems with a recursively enumerable set of axioms never occur in practice, and, furthermore, any such set of axioms can easily be replaced by recursive set of equivalent axioms. So let us assume that the set of axioms is decidable. The statement that a sentence ϕ is provable has the form:

> *there exists a sequence of formulas such that every formula in the sequence is either an axiom of logic, or an axiom of the theory T or it follows from some previous formulas by a logical rule, and the last formula in the sequence is ϕ.*

Clearly the only quantifier is *'there exists a sequence'*, the remaining part is a statement that can be decided by an algorithm. This is exactly the form of Σ_1 sentences.

Now we can finish the proof quickly. Suppose that γ_T is provable in T. Then, by Σ-completeness, also $\pi_T(\ulcorner\gamma_T\urcorner)$ is provable in T. But γ_T is equivalent to $\neg\pi_T(\ulcorner\gamma_T\urcorner)$. Hence we have both $\pi_T(\ulcorner\gamma_T\urcorner)$ and $\neg\pi_T(\ulcorner\gamma_T\urcorner)$, which means that T is inconsistent. Thus, assuming that T is consistent, γ_T is not provable in T. Furthermore, γ_T is true, because γ_T is equivalent to $\neg\pi_T(\ulcorner\gamma_T\urcorner)$ which expresses the true fact that γ_T is not provable in T.

Rosser's formula. In the exposition above I used the assumption that the theory T is true. Gödel stated his theorems in a greater generality by using the weaker assumption of *ω-consistency.* This property is the restriction of the assumption that T is true (proves only true sentences) to the class of Σ_1 sentences. Thus ω-consistency means that if ψ is provable in T, for ψ a Σ_1 sentence, then ψ is true. Equivalently we can say that if $\exists x\phi(x)$ is provable in T for a Σ_0 formula $\phi(x)$, then ϕ holds for a concrete number. (One can also view this property as the converse of Σ-completeness.) The reason for introducing this concept was not only to get the most general theorem, but also because the concept of a true theory is rather vague. Suppose we consider extensions of ZF that decide the Continuum Hypothesis. Which of these should we call true? By restricting this property to Σ_1 sentences about natural numbers he was able to avoid such ambiguities.[3]

Thus Gödel's results left open the possibility that some ω-inconsistent theories were complete. However, it is not difficult to rule out also this strange possibility. That was done by J.B. Rosser shortly after Gödel. He defined the following self-referential sentence ρ_T:

> *For every proof of ρ_T, there exists a shorter proof of $\neg\rho_T$.*

The idea behind this construction is to obtain a sentence that is symmetric in the sense that the sentence and its negation behave in the same way. The sentence ρ_T is surely not quite symmetric. I says that *"if there exists* a proof of ρ_T, then ...", whereas its negation says *"there exists* a proof of $\neg\rho_T$ ...". However we get the symmetry if we assume that *either there exists a proof of ρ_T or there exists a proof of $\neg\rho_T$.* Under this assumption ρ_T is equivalent to:

[3] If we consider it formally, we are working in some theory when proving the incompleteness theorems, say in ZF. Then by the Gödel-Tarski Theorem we cannot define the set of the true statements about sets. However, for various subsets we can define the truth, in particular, we can define the truth for arithmetical sentences.

the shortest among the proofs of ρ_T *and* $\neg\rho_T$ *is a proof of* $\neg\rho_T$*,*

and $\neg\rho_T$ is equivalent to:

the shortest among the proofs of ρ_T *and* $\neg\rho_T$ *is a proof of* ρ_T*.*

After these preliminary considerations the proof is easy. Assume that ρ_T is not independent. Well, this is exactly the assumption that guarantees the symmetry. Furthermore, according to the Σ-completeness this statement is also provable in T.

So it suffices to consider one of the two (non-exclusive) possibilities, say that $\neg\rho_T$ is provable in T. Let d be such a proof of $\neg\rho_T$. Since T is consistent, there is no proof of ρ_T in T, in particular, there is no such proof below d. The sentence *"d is a proof of $\neg\rho_T$ and there exists no proof of ρ_T below d"* is a Σ_1 sentence (in fact even Σ_0), thus, by Σ-completeness, it is provable in T. But this sentence implies ρ_T, hence T is not consistent. Thus we got a contradiction, which shows that $\neg\rho_T$ is not provable in T.

Some attempts to get round the second incompleteness theorem. Gödel's Theorem excludes the possibility for a theory to prove its consistency. To be sure that you understand this theorem and the concepts involved, it is instructive to look at some attempts to extend a given theory T so that the extension proves its consistency. In the sequel I will write $Con(T)$ instead of Con_T in order to avoid indices with indices.

1. Define $T_0 = T$, $T_1 = T_0 \cup \{Con(T_0)\}$, $T_2 = T_1 \cup \{Con(T_1)\}$ and so on. Let S be the union of theories $T_0, T_1, T_2, \ldots$. Assume that T is a true theory. Then, not only is T consistent, but also T_1, because $Con(T)$ is true. Moreover, T_1 is also true. In the same manner we can prove by induction that all T_n are true.

Here is a *wrong argument* that S proves its consistency.

> *Suppose that a contradiction can be derived from* S*. Then it must be derived from some finite part, say* T_n*. Now,* $Con(T_n)$ *is provable in* S*, because it is already provable in* T_{n+1}*.*

What is wrong in this argument? We want to prove $Con(S)$ in S. Let us see what is the sentence $Con(S)$ that we want to prove. It says that S is consistent, hence it must imply that every part of S is also consistent. Thus in S the following implication is provable $Con(S) \rightarrow \forall x\, Con(T_x)$. One can prove in S also the converse implication. In fact, if we formalize the above wrong argument we get a correct proof of $\forall x\, Con(T_x) \rightarrow Con(S)$ in S. So the problem with the above argument is that we are trying to use concrete instances $Con(T_0), Con(T_1), Con(T_2), \ldots$ instead of the sentence $\forall x\, Con(T_x)$.

As I already said before, we cannot use such a deduction, but perhaps, I should give a more formal explanation why. Let me simplify the problem and look at what is wrong with the following argument. Suppose T proves all concrete instances $\varphi(0), \varphi(1), \varphi(2), \varphi(3) \ldots$. Now we want to deduce $\forall x\, \varphi(x)$. The wrong argument is as follows.

> *Suppose* $\varphi(n)$ *is false for some* n*. Then we get a contradiction, because* $\varphi(n)$ *is provable in* T*. Hence* $\varphi(n)$ *is true for all* n*.*

Now it is more apparent that in this argument we mix concrete instances with general statements. When we say *'Suppose* $\varphi(n)$ *is false for some* n*',* we do not fix n to be any concrete number, on the contrary, n must be a generic element. In formal proofs n will be represented by a variable, whereas in $\varphi(0), \varphi(1), \varphi(2), \varphi(3) \ldots$ we shall have closed terms such as $0, 1, 1+1, 1+1+1, 1+1+1, \ldots$. In logic we can generalize

$\varphi(x)$ to $\forall x\ \varphi(x)$ (provided that x is a variable), but we cannot derive such a sentence from $\varphi(t)$ for a closed term (or from any number of such sentences).

2. Since the above construction did not produce what we wanted, one may try to construct a kind of limit of the process of adding consistencies. The Fixed Point Theorem mentioned above seems the right tool for this purpose, since usually a limit is a fixed point. So what we want is a sentence δ such that

$$\delta \equiv T \text{ with } \delta \text{ is consistent,}$$

or, using the Con notation

$$\delta \equiv Con(T \cup \{\delta\}).$$

Our hope is that $T \cup \{\delta\}$ is consistent and proves its own consistency. The second condition follows immediately from the construction of δ. Hence, by Gödel's Theorem, the other condition must fail, so $T \cup \{\delta\}$ is not consistent. The proof of the second incompleteness is not difficult, thus we can see how the contradiction appears by following its proof for this particular theory line by line, but can we give some intuitive explanation why it is so? Here is a possible one. If we add consistency statements one by one as in 1., we get stronger and stronger theories. In $T \cup \{\delta\}$ we add a sort of loop that generates automatically such extensions. Because of that the strength of the theory goes beyond any limit and thus becomes too strong—inconsistent.

3. Another process that has been studied is to extend the process of adding consistencies transfinitely. I will have to say more about it later. ♠

Arithmetization of syntax in Peano Arithmetic. Peano Arithmetic has only $+$ and $\times$ as primitives. Therefore, arithmetization of syntax, or which is is essentially the same task, interpretation of the Finite Set Theory in Peano Arithmetic, is not quite easy. The problem is that all natural ways of coding sequences of numbers by numbers are based on exponentiation which is not among primitives. Exponentiation can be defined, but again it is hard to do it without having a way of coding sequences.

Gödel devised a tricky way of coding sequences that avoids exponentiation, his *β-function:*

$$\beta(u, d, i) = \mathrm{Rem}(u, id + 1),$$

where $\mathrm{Rem}(x, y)$ denotes the remainder of x after division by y. Using this function one can code arbitrary finite sequences of natural numbers. Given a sequence $a_1, \dots, a_n$ the code is a triple (u, d, n) such that for all $i = 1, \dots, n$, $a_i = \beta(u, d, i)$. We can find such a code as follows. Take $d = (\max_i a_i n)!$, then the numbers $d+1, 2d+1, \dots, nd+1$ are pairwise relatively prime, thus, by the Chinese Remainder Theorem, there exists u such that

$$u \equiv a_i \mod (id + 1), \text{ for } i = 1, \dots, n.$$

Since $a_i < id + 1$, we have $a_i = \mathrm{Rem}(u, id + 1) = \beta(u, d, i)$. Furthermore, β-function has a simple arithmetical definition:

$$\beta(u, d, i) = a \ \equiv\ \exists z \exists w (u = (id + 1)z + a \wedge a + w = id).$$

Several other methods of coding finite sequences have been found. It is possible to first define exponentiation and then use definitions of sequences based on it.

Peano Arithmetic is surely not the weakest theory for which one can prove the incompleteness theorems. One can show that the first incompleteness theorem holds true for any consistent theory that contains only the basic seven axioms of Peano

Arithmetic, without the schema of if induction (see page **??**). The theory axiomatized by these seven axioms is called *Robinson's Arithmetic* and it is very weak. It cannot even prove such basic facts as the commutativity of addition.

Given such a weak theory it is not clear whether it makes sense at all to ask if the second incompleteness theorem holds in it. Proving the second incompleteness theorem means proving that it is consistent to assume in T that there exists a proof of contradiction in T. But if we cannot prove the basic properties of the formalized syntax, the proof of contradiction would only be some number satisfying some strange formula.

Thus it is a bit surprising that one can prove a meaningful version of the second incompleteness theorem even for such weak theories. The point is that one can define an interpretation of a much stronger theory in Robinson's Arithmetic, a theory that is not as strong as PA, but strong enough to properly formalize syntax. The interpretation is defined by a suitable formula $\nu(x)$. The numbers that satisfy ν form an initial segment of the whole universe of numbers of Robinson's Arithmetic, a segment closed under addition and multiplication. The interpretation is the restriction to ν. Then one can prove that it is consistent to assume that there exists a proof of contradiction already in the segment defined by ν. Since on the segment we have all the basic properties, it is a number that indeed encodes a proof of contradiction.

Proofs of the impossibility of impossibility. Sometimes people ask if it possible to prove some kind of higher level incompleteness in the sense that the statement of unprovability itself is unprovable. A trivial example of this kind is to consider the question whether a contradiction, say $\phi \wedge \neg\phi$ is provable in a theory T. By Gödel, T is unable to decide if it is consistent, so it is unable to decide if $\phi \wedge \neg\phi$ is provable in T or not. To get something a little bit more interesting, suppose we want to have theories S and T and a sentence ϕ such that

1. it is provable in S that T is consistent;
2. it is consistent with S to assume that ϕ is provable in T;
3. it is consistent with S to assume that ϕ is *not* provable in T.

Again, the existence of such theories and such a sentence is a straightforward consequence of the second incompleteness theorem. Take T some theory that is strong enough for Gödel's Theorem to be applicable. Let S be $T+Con(T)$, thus 1. is satisfied. Let ϕ be $\neg Con(T)$. By Gödel's Theorem, we have both S is consistent with $\neg Con(S)$ and S is consistent with $Con(S)$. Note that $\neg Con(S)$ is $\neg Con(T + Con(T))$, which means that T proves $\neg Con(T)$ which is ϕ. Thus we have 2. Furthermore, $Con(S)$ is $Con(T + Con(T))$, which means that $\neg Con(T)$ is not provable in T, thus we get also 3.

A more interesting question is the following. Is it possible that we prove that ϕ is undecidable in some theory without knowing, if ϕ is true or false? The best example is the Continuum Hypothesis (which I will deal with in the last section of this chapter). It has been shown independent from Zermelo-Fraenkel set theory and up to now no axiom nor even convincing argument was found that would decide it. But this kind of independence is not possible for all sentences of all types. The most interesting ones are Π_1 sentences because they are universal statements about numbers or about finite sets. They include statements that a certain equation is unsolvable, say, that $x^3 + y^3 = z^3$ does not have a solution with natural numbers x, y, z greater than 0. If we construct a model in which the equation is unsolvable, then it is unsolvable, because the actual natural numbers are contained in any model of arithmetic. On the other hand, finding a model where such an equation does have a solution, may have no implications.

Hence the least arithmetical complexity for which such independence can occur is Π_2 and Σ_2. In a short unpublished note Petr Hájek has described a contrived example of such an independence result. Here is a brief sketch of a version of his result. Let $P_{PA}(x)$ resp. $P_{ZF}(x)$ denote formulas expressing the provability predicates of Peano Arithmetic resp. Zermelo-Fraenkel set theory. Applying the Diagonal Lemma construct a sentence δ such that PA proves

$$\delta \equiv P_{ZF}(\ulcorner\delta\urcorner) \to P_{PA}(\ulcorner\neg\delta\urcorner). \tag{4}$$

Since ZF proves all (translations of) theorems of PA, the sentence above is also provable in ZF. We shall show that

1. it is provable in ZF that δ is independent from PA and
2. δ is independent from ZF.

For the second part of the statement we need to assume that ZF is consistent and does not prove that it is inconsistent; needless to say, these assumptions are accepted as true by most mathematicians.

First we shall argue in ZF and prove the independence from PA. We shall use the fact that for every arithmetical formula ψ we can prove in ZF

$$P_{PA}(\ulcorner\psi\urcorner) \to \psi$$

This is called the *reflection principle for* PA and it is an assumption stronger than the statement that PA is consistent.

Still arguing in ZF, suppose $PA \vdash \delta$, then we get $PA \vdash P_{PA}(\ulcorner\delta\urcorner)$, by Σ-completeness, and also $PA \vdash P_{ZF}(\ulcorner\delta\urcorner)$, as PA can see that ZF is at least as strong as PA. From $PA \vdash \delta$ and $PA \vdash P_{ZF}(\ulcorner\delta\urcorner)$ we get $PA \vdash P_{PA}(\ulcorner\neg\delta\urcorner)$ using the definition of δ and modus ponens. Thus PA proves that it is inconsistent. By the reflection principle, we have also that PA is inconsistent, but in ZF we know that it is consistent. Thus PA does not prove δ.

Now suppose $PA \vdash \neg\delta$. Then $PA \vdash P_{PA}(\ulcorner\neg\delta\urcorner)$. Note that $\neg\delta$ is $P_{ZF}(\ulcorner\delta\urcorner) \wedge \neg P_{PA}(\ulcorner\neg\delta\urcorner)$, so we have also $PA \vdash \neg P_{PA}(\ulcorner\delta\urcorner)$. Thus PA is inconsistent, which is not true. Hence we have shown in ZF that δ is independent from PA.

Now we shall prove that δ is independent from ZF.

Suppose $ZF \vdash \delta$. Then $ZF \vdash P_{ZF}(\ulcorner\delta\urcorner)$, hence $ZF \vdash P_{PA}(\ulcorner\neg\delta\urcorner)$. By reflection, $ZF \vdash \neg\delta$, thus ZF would be inconsistent. So $ZF \nvdash \delta$.

Suppose $ZF \vdash \neg\delta$. Then $ZF \vdash P_{ZF}(\ulcorner\neg\delta\urcorner)$ and $ZF \vdash P_{ZF}(\ulcorner\delta\urcorner) \wedge \neg P_{PA}(\ulcorner\delta\urcorner)$, by the definition of δ, see (4). Whence $ZF \vdash P_{ZF}(\ulcorner\neg\delta\urcorner) \wedge P_{ZF}(\ulcorner\delta\urcorner)$. Thus ZF proves that it is not consistent, which is not true. Hence δ is independent from ZF.

Modal logic of provability. Let $P_{PA}(x)$ denote, as above, that the formula with the Gödel number x is provable in PA. In particular it defines an operator that from a given formula ϕ produces another formula $P_{PA}(\ulcorner\phi\urcorner)$. It is natural to think of this operator as a modality, as the meaning of $P_{PA}(\ulcorner\phi\urcorner)$ is that ϕ holds in some stronger sense, namely we have a proof of it. So let us use $\Box$ for it and let us see what kind of modal logic it gives. Again, we will only be interested in the propositional fragment of such a logic. I will speak only about the provability in PA, but the results are valid for a large class of theories. Nevertheless, you should keep in mind that the properties of this modality may depend on the theory whose provability we consider.

It has been shown that this gives rise to a modal logic which can be axiomatized by a few simple axiom schemata and rules. This logic is called *Provability Logic.* It is not very difficult to prove that the provability predicate for PA satisfies all axioms and rules of the Provability Logic. The difficult part of the characterization is to show that all true statements about provability in PA expressible in modal logic can be proved in the Provability Logic. This was proved by R. Solovay in 1976.

I will start with a little different system and prove Gödel's Second Incompleteness Theorem in it. The system is axiomatized by the standard axioms and rules of the classical propositional calculus and the following modal ones: the generalization rule

(1) *from* ϕ, *derive* $\Box\phi$,

two axiom schemas

(2) $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$,

(3) $\Box\phi \to \Box\Box\phi$,

and a schema that formalizes the Diagonal Lemma

(4) *for every formula* $\phi(x)$ *in which the propositional variable* x *occurs only in the scope of the modality* $\Box$, *one may introduce a new propositional constant* d *and add* $d \equiv \phi(d)$ *as a new axiom.*

The conditions (1),(2),(3) are satisfied by the usual provability predicate of Peano Arithmetic. They are also called *the derivability conditions*, as they axiomatize which properties the provability predicate must satisfy so that we can prove Gödel's Theorem. The Diagonal Lemma is not included in the derivability conditions, because it applies to all formulas. In this book I have always assumed that the provability predicate is formalized in a natural way. This means that we express the syntactical concepts needed in the formalization in the most direct way. However, it is natural to ask for which other formalizations Gödel's Theorem still holds true. One possible answer to this question is that these are all predicates that satisfy conditions (1), (2) and (3) above.

So let us prove Gödel's theorems in this formalism. By (4) we get a constant c and the sentence

$$c \equiv \neg\Box c \tag{5}$$

Then, in particular, we have

$$c \to \neg\Box c.$$

Applying generalization (1), axiom (2) and modus ponens we get

$$\Box c \to \Box\neg\Box c.$$

By (3) we also have

$$\Box c \to \Box\Box c.$$

The last two give together

$$\Box c \to \Box\Box c \wedge \Box\neg\Box c.$$

Using classical rules and (2) one can easily show $\Box\Box c \wedge \Box\neg\Box c \to \Box(\Box c \wedge \neg\Box c)$ which gives

$$\Box c \to \Box(\Box c \wedge \neg\Box c).$$

Let us abbreviate $\Box c$ by ϕ. Thus the last formula is

$$\Box c \to \Box(\phi \wedge \neg\phi)$$

Now the meaning of the consequent is clear, it says that a contradiction is provable. (The contradiction is represented by $\phi \wedge \neg\phi$, but we know that all contradictions are equivalent.) Let us take the contrapositive implication

$$\neg\Box(\phi \wedge \neg\phi) \to \neg\Box c. \tag{6}$$

The meaning is that if a contradiction is not provable then c is not provable. But if c is not provable then, by (5), c is true. Thus we have obtained the First Incompleteness Theorem. The Second Incompleteness Theorem follows immediately by observing that the meaning of $\neg\Box(\phi \wedge \neg\phi)$ is the consistency of PA and that it implies c.

The above system is not a version of modal logic, as it needs additional constants. It is interesting that it is possible to avoid this problem by replacing the diagonal schema (4) with one simple axiom schema:

(5) $\Box(\Box\phi \to \phi) \to \Box\phi$.

In some sense, this axiom is the opposite of the axiom $\Box\phi \to \phi$ of S_4, which is not valid in PRL. It says, in terms of provability in PA that the only way one can prove $P_{PA}(\ulcorner\phi\urcorner) \to \phi$ is to prove ϕ. This statement is known as *Löb's Theorem*, thus the axiom is also called *Löb's axiom*. The logic axiomatized by (1),(2),(3),(5) is the Provability Logic PRL.

The equivalence of these two systems is based on a difficult theorem, first proved by G. Sambin and D. de Jongh. The theorem says that in PRL one can define explicitly all fixed points. This means that given a modal formula $\phi(x)$ in which the propositional variable x occurs only in the scope of the modality $\Box$, one can find a modal formula ψ such that $\psi \equiv \phi(\psi)$ is provable in PRL. In particular, you can show as an exercise that for every α, the formula $\neg\Box(\alpha \wedge \neg\alpha)$ (a formula expressing the consistency) is a fixed point of $\neg\Box x$ (the formula used in the proof of Gödel's theorems). This proves that PRL is at least as strong as the system with the diagonal schema (4). The opposite direction is easy. One needs only to prove Löb's theorem in the system with diagonalization which is of about the same degree of difficulty as the proof of the incompleteness theorems that I showed above.

Does this mean that we can prove Gödel's theorems without diagonalization? No, because we would still need to prove Löb's Theorem and the only way to prove it that we know of is, again, diagonalization.

A definition of a provability predicate for which Gödel's Theorem fails. Let T be an arbitrary recursively axiomatized theory (I do not exclude inconsistent theories). Let $P_T(x)$ be the natural definition of the provability predicate for T. Define

$$P^*_T(x) \;\equiv\; P_T(x) \wedge Con(T)$$

Then we can prove $\neg P^*_T(\ulcorner 0 = 1\urcorner)$, the consistency expressed using P^*_T, without knowing almost anything about T. The argument is: either $Con(T)$ is true and then we cannot derive a contradiction from the axioms of T, or it is false, in which case the provability predicate $P^*_T(x)$ defines an empty set.

If T is consistent, then $P_T(x)$ and $P^*_T(x)$ define the same set of sentences provable in T. Thus, for example, P^*_{PA} defines (in $\mathbb{N}$) the sentences provable in Peano Arithmetic, and Peano Arithmetic proves $\neg P^*_{PA}(\ulcorner 0 = 1\urcorner)$.

This definition is, clearly, not natural. In particular it has higher quantifier complexity than is needed and it does not satisfy the provability condition (3).

A decreasing sequence of theories. R. Solovay, C. Smoryński and H. Friedman showed that it is possible to construct a sequence of theories with decreasing consistency strength, a sequence of recursively axiomatized theories $S_0, S_1, S_2, \ldots$ such that S_n proves the consistency of S_{n+1} and every S_n is consistent.

The basic idea can be explained as follows. Take an *increasing sequence* of consistent recursively axiomatized theories $T_0, T_1, T_2, \ldots$; so T_{n+1} proves the consistency of T_n. We know how to construct such a sequence—define T_{n+1} as $T \cup \{Con(T_n)\}$. Now consider a nonstandard model of arithmetic M. In this model we have theories T_ν also for infinitely large numbers ν. Thus in M we can take an infinite number ν and define $S_n = T_{\nu-n}$ for *all finite* n. What remains to do is to simulate this construction without referring to a nonstandard model. The trick is that, due to the incompleteness theorem, for every recursively axiomatized extension of Peano Arithmetic T we can define a number which either does not exist or is infinitely large, and it is consistent with T that the number exists. Namely, take ν to be the least number that encodes a proof of contradiction in T. For our construction T will be the theory $T_0 \cup \{\forall x\ Con(T_x)\}$ and we define $S_n = T_{\nu-n}$ if ν exists, otherwise $S_n = T_0$.

Let us prove that the consistency of S_{n+1} is provable in S_n. We argue in S_n as follows. Either T is consistent, then in particular T_0 is consistent and $S_{n+1} = T_0$, so S_{n+1} is consistent. Or there exists ν and then $S_n = S_{n+1} \cup \{Con(S_{n+1})\}$, hence we have $Con(S_{n+1})$ as an axiom. Let us observe that the theories S_n are not only consistent, but they are also true (they prove only sentences valid in $\mathbb{N}$).

Why is it an important result? Let us define a strict quasiordering on the set of recursively axiomatizable theories by putting $S < T$ if T proves the Π_2-reflection principle of S. (The relation is transitive, because the Π_2-reflection principle of a theory is a Π_2 sentence.) If there were no sequence of recursively axiomatized consistent theories $S_0, S_1, S_2, \ldots$ such that S_n proves the Π_2-reflection principle of S_{n+1}, then the quasiordering would be well-founded. Hence it would be possible to assign an ordinal number to every such theory so that the theories higher in the ordering would get larger ordinals. Thus it would be possible to measure the strength of theories by ordinals. However, this is not possible, because the same construction as above works with the consistency replaced by the Π_2-reflection principle.