1 Aims and Ends

Philosophers wonder whether some truths are not verifiable and whether there are things that are synthetic but a priori. They discuss whether all mathematical truths are logically valid and whether moral claims can be true. Many agree that whatever is known is true, that there are unknown justified true beliefs, and that whatever is logically provable is analytic. Logicians maintain that there are unprovable truths and that the claim that no contradiction is provable is not provable.

The reader will recognize some of the most famous problems in philosophy in this list. They are often existential or universal claims involving truth, necessity, logical validity, formal or absolute provability, analyticity, verifiability, apriority, knowability, and so on. Of course, one would expect that the respective discussions require arguments specific to the claim in question. However, all these discussions are threatened by what are usually called the *semantic paradoxes*.[1] The best known of these paradoxes is the liar paradox. It has many versions; here is one of them:

> *The sentence in italics on this page is not true.*

The reader will be familiar with the reasoning that leads to a contradiction. Assuming that the sentence is true leads to a contradiction, because it says that it is not true. Therefore, it is *not* true. This is what the sentence is saying, and thus the sentence *is* true. But that has been ruled out already. This is the liar paradox in a nutshell.

There are many other paradoxes involving truth, each with its own variants, refinements and formalizations. Notions other than truth are affected as well.

[1] The label 'semantic paradox' is problematic for various reasons, and we just use it for the kind of paradoxes that are introduced in this chapter by example. This distinction between semantic and logical paradoxes goes back at least to Ramsey (1926). In chapter 4 we show how closely they are related and that Russell's paradox can easily be converted into a semantic paradox. We have no ambition to provide a classification of paradoxes here, and for our purposes there is no need to be more precise.

As an example we choose, somewhat randomly, verifiability and a paradox very close to the liar paradox:

(V) (V) is not verifiable.

The point of labelling the sentence with '(V)' is that we have the following identity:

(V) = '(V) is not verifiable'.

Verifiability can be understood in different ways here, for instance, in the sense of 'can be demonstrated to be true', as 'can be known', or as 'is provable'. For this paradox we give the steps leading to a contradiction:

1. Assume (V) is verifiable.
2. Then '(V) is not verifiable' is verifiable (by the identity above).
3. Therefore, (V) is not verifiable (because this can be verified).
4. Hence, (V) is not verifiable (because the first line and the previous line, derived from it, contradict each other, and thus the assumption in the first line is refuted).
5. That is, we have just verified (V) (because the preceding sentence is just (V)).

The last two claims are contradictory.

The reasoning is by any means not unassailable, and various steps are questionable. However, the argument shows that it is easy to become entangled in the paradoxes. If we are not careful, we can, without very extravagant assumptions about truth or verifiability, arrive at contradictions. Any further agonizing about realism in the form of the claim that there may be truths that are not verifiable is pointless, if basic assumptions about the fundamental concepts in the debate can be used to derive a contradiction.

The reader may object at this point that we should have reasoned in a logic which does not support one of the steps, that we should not have ascribed verifiability to sentences but rather propositions, that some assumption on verifiability implicit in the reasoning ought to be rejected, or that something is wrong with labelling a sentence with a label that is already used in the sentence. For the time being, we only claim that the threat of the paradoxes should be taken seriously. If one of the mentioned objections is justified, it does have consequences for the further discussion of all the philosophical claims. If classical logic is to be rejected, we have to do metaphysics, epistemology, and so on in a nonclassical logic. If sentences cannot be verified, but, say, only propositions conceived as sets of possible worlds, then we need to reject as verifications proofs as lists of sentences. If we give up some fundamental assumption about verifiability, for instance, the assumption implicit in the transition from 2. to 3., we need to purge this assumption, as plausible and unproblematic as it may look, from all later reasoning about verifiability. If we disallow the use of labels such as (V) and try to block the derivation of the paradox in this way, still other assumptions that have a similar effect as our way of labelling (V) need to be revised; we have to give up basic reasoning about strings of symbols.

At any rate, philosophers need to think about the semantic paradoxes. As long as they have no strategy for blocking the derivation of the contradiction (or defusing it), all discussions about the claims above are up in the air.

1.1 The Quick Road to Paradox

Instead of using descriptions such as 'the sentence in italics', other constructions can be employed to obtain a liar sentence. For instance, we can use labels:

(L) (L) is not true.

Other versions, such as the following liar sentence, rely on pronouns:

This sentence is not true.

The first reaction to these and similar versions of the liar paradoxes is often a suspicion that something has gone wrong with the labelling of the sentence and the use of the pronoun 'this'. Thus, the blame for the paradox would go to the strange use of the label '(L)' and the pronoun, not to the truth predicate 'is true' or assumptions about the behaviour of the truth predicate. Avoiding this kind of use of labels and pronouns may be possible.

However, the liar paradox can be generated using purely syntactic means, without any pronouns or labels as in the mentioned examples. We use a variant of an example by Quine (1976b). The quotation of an expression is the expression enclosed in quotation marks. Now a liar sentence can be stated in the following way:

> 'preceded by its own quotation is not true' preceded by its own quotation is not true.

The means used in this variant of the liar paradox are far less dispensable. Enclosing an expression in quotation marks and appending an expression to another expression are very basic syntactic operations. Thus, trying to solve the paradoxes by restricting the means used in this version of the liar paradox is much harder; it would mean to sacrifice our ability to talk about expressions in any meaningful way.

However, we will not employ Quine's elegant trick, but a more versatile method due to Gödel. His fundamental result which permits to construct a sentence that is equivalent to the claim that it itself is not true (or provable or verifiable) is the so-called diagonal lemma.[2]

Our way to the paradoxes is quick in the sense that it permits a proof of the diagonal lemma and thus the formulation of the liar paradox in a very simple formal theory of syntax. The theory is versatile enough to permit the formulation of more sophisticated paradoxes such as Visser's and Yablo's. The theory can be used as a general framework for studying semantic paradoxes. In chapter 6 a collection of paradoxes will be presented, and a unified analysis of them is given in chapter 7. In this way we provide the reader with an access to the theory of the paradoxes that is quick and easy, uses only the means of syntax, and is formally precise. We use syntax theory – rather than labels such as '(L)' or pronouns as above – because the methods used in syntax theory are also those used (for good reason) in the more technical literature. The more sophisticated ones are best presented in syntax theory, as are the offspring of the paradoxes, such as Tarski's theorem on the undefinability of truth and Gödel's incompleteness theorems.

Our way of presenting the paradoxes can bridge the gap to the more technical literature on them. We hope that this is useful, because the technical apparatus that is applied to understand, solve, or analyze the paradoxes has become so sophisticated that only a relatively small group of logicians and technically versed philosophers have followed the rapid development of the theory of paradoxes.

It is an unfortunate consequence of this development that philosophers have often opted to leave the difficulties caused by the paradoxes aside and to concentrate exclusively on the 'philosophical' problems of the modal notions. We

[2] The diagonal lemma is actually a family of results with several variants and generalizations. We ascribe the diagonal lemma to Gödel, although he did not state it in its general form in his 1931.

think that it is not a promising strategy to devise theories of truth, necessity, and other modal notions while ignoring the paradoxes and hoping that they can be solved independently from the chosen philosophical account of the modal notions. Paradoxes are not just a nuisance that gets in the way of philosophical theorizing. They can also be the engine behind progress. In almost any area of philosophy, paradoxes of some kind have prompted new approaches. Paradoxes that arise from purely syntactic considerations can, and actually did, prompt philosophers of truth to rethink their theories. The situation may be compared to set theory, where Russell's and related paradoxes led to the formulation of modern axiomatic theories of sets; it may even be compared to the paradoxes that led Einstein to devise the theory of special relativity.

One reason why the literature on the paradoxes has become difficult to access is that they are often not studied before the background of a syntax theory. Instead they are studied before the background of a mathematical theory, usually arithmetic. The syntactic objects, that is, strings of symbols, including sentences, are then assigned numbers as codes. Syntactic operations, such as concatenation, substitution of symbols, etc., are then mimicked in the arithmetical language, using operations on these codes. That the theory of syntax can be simulated in arithmetic is one of Gödel's (1931) central insights. We think it is fair to say that, before Gödel, logicians assumed that this could be done in some way. However, it took Gödel to work out a precise mathematical account, which involves some results from number and computability theory. If one aims to show that some system of arithmetic is incomplete, as Gödel did, these results are hard to avoid. However, if one is interested in the analysis of the paradoxes and in a logical framework for studying modal notions, then it is not clear why the theory of syntax has to be transposed into arithmetic. In fact, for many applications in philosophy it is far more natural to study the paradoxes before the background of a theory of syntax and to bypass arithmetic entirely.

In this book, we present formal theories of syntax which capture our informal reasoning about syntax. This will greatly simplify the presentation of the paradoxes, but also of Gödel's incompleteness theorems. We hope that this will make the material more accessible and attractive to readers without a mathematical background.

Of course, for some goals there are very good reasons for taking the detour via arithmetic. For the purposes of an article in a logic journal it is much easier to state at the beginning that some system of arithmetic will be used as a proxy for a syntax theory, usually without specifying any specific coding. The specialist reader does not need reminding of the underlying assumptions about coding. Moreover, using arithmetic as framework permits one to tap into the wealth of existing results about such theories.

Many axiomatizations of the theory of syntax have been given in the literature. Some are intended to be theories of syntax for some object language, not necessarily the language of syntax theory itself. Some obey nominalistic constraints that are compatible with the assumption that there are only finitely many expressions. Others axiomatize a theory of syntax for a language with only two symbols, so that coding is required for languages in normal notation. As far as we know, Tarski (1935, § 2) was the first to axiomatize syntax theory. He used concatenation as a primitive operator and formulated his theory in second-order logic. Our system in this chapter employs some axioms found in Tarski's paper. His system has been developed, expanded, and analyzed by Corcoran, Frank, and Maloney (1974) and other logicians. Another classic in the field is Quine's (1940, chapter 7) protosyntax. He also developed a strictly nominalistic theory in (Goodman and Quine 1947). Smullyan (1957) provided a proof of the diagonal lemma in a very simple and elegant axiomatic theory. Grzegorczyk's (2005) proof of the undecidability of predicate logic in syntax theory (see also Visser 2009) contains many ideas that are mainly used in our more expressive theory in chapter 8. However, it differs significantly from our approach in axiomatizing the syntax of a language that has only two distinct symbols, so that, as mentioned above, coding is still required for languages in a normal notation. Blau (2008) uses a theory of quotation as his basis for analyzing the paradoxes. Perlis (1988) presents the paradoxes in a style similar to ours.

These approaches have influenced ours, which may well not contain any fundamentally new ideas. We have not tried to trace back our axioms and proofs to earlier versions in the literature. Many are straightforward formalizations of metatheoretic principles frequently used in reasoning about formal languages; the metatheoretic versions are commonly considered obvious, and the contribution of the logicians just mentioned is mainly to have made these principles formally explicit. It is not unlikely that many authors have arrived at similar principles independently. In many cases axioms and claims differ between various accounts in details; for instance, the empty string can be admitted or not. All this often makes it hard to identify a particular statement or observation in the literature as the original source of an axiom or theorem in syntax theory.

In contrast to many other authors, we do not aim at an *elegant* theory of syntax; our theory is aimed at capturing more directly our informal theory of syntax, even if that proves to be more clumsy than streamlined theories such as Grzegorczyk's. Our austere notion of quotation is that used in logical metatheory. In our syntax theories we do not attempt to capture the general phenomenon of quotation outside this confined range of application. For puzzles and observations arising from quotation elsewhere we refer the reader to (Cappelen and Lepore 2007).

1.2 The Direct Way to Paradox

This may sound as if our theories of syntax served merely the purpose to provide an easier route to theories of truth and the incompleteness theorems, avoiding coding and arithmetization. However, it also serves an important theoretical purpose.

A theory of syntax is always required, even in the usual proofs of the incompleteness theorems, which proceed via coding. In order to theorize about the relation between syntactic and arithmetical notions, we need a sufficiently precise theory of syntax. When we claim that the concept of being a sentence of the chosen language or of being provable in a specific system can be expressed in arithmetic, we should better be clear about these concepts.

Usually, the theory of syntax is not fully formalized in proofs of the incompleteness theorems. We do not think that something is wrong about this approach, but having a formal theory of syntax makes assumptions explicit that are usually left implicit in the metatheory. The reader familiar with one of the traditional proofs of the second incompleteness theorem or even the discussions about intensionality in metamathematics, as discussed by Feferman (1960) and many subsequent authors, will have seen claims that a certain formula of an arithmetical theory 'naturally expresses' provability. It is notoriously difficult to spell out what this means, but it must be some structural similarity between a formula in arithmetic and the definition of provability in informal metatheoretic syntax theory. If one aims to make the notion of natural expression (and similar concepts) precise, it would be useful to have a fully explicit and formal syntax theory like our theory E^*. Some observations about problems of intensionality in metamathematics and the use of syntax theories in their analysis will be discussed in the final chapter 12.

Cambridge University Press & Assessment
978-1-108-84101-6 — The Road to Paradox
Volker Halbach , Graham E. Leigh
Excerpt
More Information

2 Technical Preliminaries

In this chapter we briefly review some basics of first-order predicate logic, which will be familiar to the expert reader. In particular, we fix our notation and review some topics, including function symbols and many-sorted languages, that are not always covered in introductory textbooks such as (Halbach 2010).

2.1 Languages of First-Order Predicate Logic

We describe the formation rules for the formulæ and sentences of our languages. All the languages we consider have an infinite stock of variables: v_0, v_1, v_2, v_3, … Some contain also function symbols. Each function symbol has an arity assigned, which is some natural number 0, 1, …

All variables are terms. If f is a function symbol of arity n and $t_1, \dots, t_n$ are terms, then $ft_1 \dots t_n$ is a term. For some binary function symbols, that is, function symbols f of arity 2, we often write $(t_1 f t_2)$ for ft_1t_2, that is, we write the function symbol between terms and use brackets around this expression; but this is only our notation for the official ft_1t_2. In mathematics the symbols for addition and multiplication are binary function symbols. They correspond to the English phrases 'the sum of … and …' and 'the product of … and …' In mathematics we write + and × between terms, for instance, $a + b$ rather than $+ab$; and this is why we adopt our notation for binary function symbols. It is also clear that in our notation brackets are required: $(a \times b) + c$ and $a \times (b + c)$, for instance, are clearly not equivalent. In the official notation brackets are not needed.

Unary function symbols correspond to expressions such as 'the mass of …' in English. The expression 'the mass of …' has arity 1. It takes a term, for instance, 'the Earth', and yields a new term, 'the mass of the Earth'. An example of a binary function expression is 'the border between … and …'

We also allow function symbols with arity 0. They are individual constants. Since individual constants are 0-place function symbols, they need to be combined with 0 many terms to yield a term, that is, they are terms already by themselves.

A language may have no function symbols at all or only individual constants, that is, function symbols of arity 0. All the languages we consider will have only finitely many function symbols.

In contrast, all our languages have at least one predicate symbol. As is the case with function symbols, each of the languages we consider will feature only a finite number of predicate symbols. Predicate symbols also come with an arity. If P is a predicate symbol with arity n and $t_1, \dots, t_n$ are terms, then $Pt_1 \dots t_n$ is an atomic formula. An exception is the binary predicate symbol $=$ for identity, which is written between terms. We use $=$ as a predicate symbol in our formal languages, but also as the usual identity symbol in the language we use, the 'metalanguage'. The identity symbol will be a predicate symbol in all of the languages we consider.

As in the case with function symbols, the predicate symbols will be specified for each language we discuss. In many elementary logic textbooks such as (Halbach 2010) a single language with infinitely many predicate symbols of arbitrary arities is considered. This ascertains that we never run out of predicate symbols when formalizing arguments in natural language. But once we consider a specific set of assumptions, that is, a theory, this will often be formulated with very few predicate symbols. Set theory, for instance, in which all (or perhaps almost all) of mathematics can be carried out, has only a single, binary predicate symbol.

All atomic formulæ are formulæ. If φ, ψ are formulæ and x is a variable, then $\neg\varphi$, $(\varphi \to \psi)$, and $\forall x\varphi$ are formulæ. We use $(\varphi \wedge \psi)$ as an abbreviation for $\neg(\varphi \to \neg\psi)$, $(\varphi \vee \psi)$ as an abbreviation for $(\neg\varphi \to \psi)$, $(\varphi \leftrightarrow \psi)$ as an abbreviation for $\neg\big((\varphi \to \psi) \to \neg(\psi \to \varphi)\big)$, and finally $\exists x\varphi$ as an abbreviation for $\neg\forall x\neg\varphi$. These abbreviations are metalinguistic abbreviations. That is, we do not introduce new symbols $\wedge$, $\vee$, $\leftrightarrow$, and $\exists$ into our formal object language; rather, we use $\wedge$, $\vee$, $\leftrightarrow$, and $\exists$ in our informal metalanguage in order to save some space and make our text more readable. As we have already done here, we use lower-case Greek letters as metavariables for formulæ. Predicates of arity 0 are also admissible and called sentence parameters.

The notion of free and bound occurrences is defined in the usual way: In an atomic formula all occurrences of variables are free. All occurrences of variables that are free in φ and in ψ are also free in $\neg\varphi$ and $(\varphi \to \psi)$. Every free occurrence of a variable y in φ different from x itself is also free in $\forall x\varphi$; all occurrences of x in $\forall x\varphi$ are bound. An occurrence of a variable is bound iff it is not free.[1] Formulæ without any free occurrences of variables are sentences.

[1] As usual, 'iff' abbreviates 'if and only if'.

We use metatheoretic rules for omitting brackets: Our formulæ are always the formulæ with all the brackets. If we omit brackets, we obtain abbreviations of formulæ. The rules for omitting brackets are specific to the particular occurrence of a formula; so they should be formulated for occurrences of formulæ. We can omit the outer brackets in an occurrence of a formula $(\varphi \rightarrow \psi)$, as long as this occurrence is not within an occurrence of another formula. That is, we can drop the 'outermost' set of brackets. The expression $(\varphi \wedge \psi \wedge \chi)$ is short for $\big((\varphi \wedge \psi) \wedge \chi)\big)$. An analogous rule applies to $\vee$. Of course, here we are already using the abbreviations $\wedge$ and $\vee$. Moreover, in abbreviated formulæ with $\wedge$ and $\vee$, $\wedge$ and $\vee$ bind more strongly than $\rightarrow$ and $\leftrightarrow$. Thus, $\varphi \wedge \psi \rightarrow \chi$, for instance, is short for $(\varphi \wedge \psi) \rightarrow \chi$, which abbreviates $\big((\varphi \wedge \psi) \rightarrow \chi\big)$.

Above we have already used x and y as metavariables for variables in the object language. This means that x could be any of the variables $\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \ldots$ As in the case of the metavariables for formulæ, we employed metavariables for variables in order to make our definitions sufficiently general. For instance, we stipulated that $\forall x \varphi$ is a formula if x is a variable and φ is a formula. It would not suffice to say that $\forall \mathsf{v}_0 \varphi$ is a formula if φ is, because this would not imply that $\forall \mathsf{v}_1 \varphi$ is a formula. In what follows we continue to use x, y, and so on as metavariables for variables.

Writing 'v_0', 'v_1', and so on makes the notation somewhat cluttered. To avoid the indices, we write x for v_0, y for v_1, z for v_2, and w for v_3. So sans-serif letters stand for specific variables, while letters in italics are metavariables for variables. In logic it usually does not matter which variable is used, as long as the variables employed in a formula are pairwise distinct. But in a theory of expressions and in proving results such as the diagonal theorem, we have to be very specific and will use sans-serif letters, that is, specific variables.

2.2 Logical Calculi

There are numerous ways to generate the logically valid sentences of the languages described in the previous section. In many textbooks some variant of the system of natural deduction is used, while tableaux systems are used in many others. There are also sequent and axiomatic systems, and calculi less likely to be seen in introductory texts for philosophers. For most parts of this book it does not matter which logical calculus is used. However, the reader should be familiar with at least one such calculus. When we prove a sentence of our formal language,