The Meaning of Mathematics

On the Meaning of Mathematical Statements in a Structuralist Context

Universiteit Utrecht

Timo Diedering (6267556)
Supervisor: dr. J. Korbmacher
Second reader: dr. E. Stei

Bachelor thesis (15 ECTS)
BA Philosophy
Faculty of Humanities
Department of Philosophy and Religious Studies
21st February 2021

Abstract

The aim of this thesis is to provide an account of the meaning of second-order formulae in a structuralist context. We first give the necessary background in notions of formality in logic and argue for the use of second-order logic as language of mathematics, as proposed by Shapiro. We then formalize Giovannini's and Schiemer's theory of structural definitions. Afterwards, we argue that the meaning of a second-order formula in a structuralist context is the isomorphism class of structures that satisfy its propositional function. Finally, we show how this theory of meaning works with respect to the structure of the natural numbers.

Contents

Introduction

In this thesis, we will be concerned with the meaning of mathematical statements. Specifically, we will answer the question: *what is the meaning of a second-order formula in a structuralist, mathematical context?* We will answer this question from the perspective of structuralism, a contemporary trend in the philosophy of mathematics that states that mathematics is about structures. From a structuralist perspective, this also answers the question what the meaning of mathematical statements is, since these can be expressed in second-order logic. In order to do so, we will first give a short historical overview.

In the philosophy of mathematics, we try to find ways to make sense of the objects that are studied by mathematicians, such as functions and sets. Another kind of such objects are numbers. What do we refer to when we reason that 2 is smaller than 3, what *is* such an object, ontologically speaking? An important question in the philosophy of mathematics that tries to give an account of how we can ground this mathematical practice is that of the *foundations of mathematics*.[1] These groundings can answer the ontological questions we have regarding mathematical objects.

The subdiscipline of the philosophy of mathematics that we call *mathematical logic* tries to formalize the way in which mathematicians reason. Although the reasoning that is done by mathematicians is intuitively plausible, it is mostly done informally. The branch of philosophy of mathematics that tries to formalize this reasoning usually tries to give a foundation in terms of the study of valid reasoning: logic.[2] There have been quite some attempts in history of giving such a foundation for mathematics. One of these is logicism, which tried to ground all of mathematics in (deductive) logic.[3] Today it is often believed that *grounding* all of mathematics in logic is a bridge too far.[4] However, there is still being argued that logic can be a good way to formalize parts of mathematics and making precise what is going on while reasoning in mathematics.[5] Even though not everything will be grounded by logic, it can be used as a formalization tool.

One of the debates in this project of formalizing mathematics in logic, is about in *what logic* one should work. The choice mainly is between first- and second-order logic. Second-order logic has more expressive power, but it has a major drawback as it is incomplete. We will discuss elaborately what this means. It does have a major asset as well, in the form of categoricity. To understand why categoricity is important, we need to discuss structuralism. Structuralism can be seen as an answer to both the question of the ontological status of mathematical objects, and the question of how mathematical reasoning can be formalized. The choice for second-order logic is natural from a structuralist perspective. Moreover, it gives us an answer of what mathematicians actually study: structures.

Dedekind is often mentioned as the first author to have started the debate on structuralism.[6] In his 1893 book *Was sind und was sollen die Zahlen?*[7] Dedekind gives some remarks that are often interpreted as being of a structuralist nature.[8] To understand well what structuralism is, however, we will look at Benacerraf's more modern contribution to the field of mathematical structuralism.

In the 1960s Benacerraf added a new account to the modern discussion of what is at issue

1. Horsten, 'Philosophy of Mathematics', 1.
2. Väänänen, 'Second-Order Logic and Foundations of Mathematics', 4.
3. Tennant, 'Logicism and Neologicism'.
4. Shapiro, *Foundations without Foundationalism*, 37.
5. Shapiro.
6. Reck, 'Dedekind's Structuralism'.
7. Dedekind, *Was sind und was sollen die Zahlen?*
8. For more information on the historical development of structuralism, please refer to Reck, 'Dedekind's Structuralism' and Reck and Schiemer, 'Structuralism in the Philosophy of Mathematics'

in mathematics in his article *What Numbers Could Not Be.*[9] He argued that, for example, the natural numbers should be thought of as a *structure* and the properties of natural numbers should be thought of as relations between elements of the structure. Whether a structure exemplifies the natural numbers is not dependent on the elements, but on the relations between them.[10] That way, we can speak of *abstract structures* that cover the relations such a structure should bear in order to be an example of the abstract structure. We focus on the structuralism as discussed in Shapiro's work.[11] Others have also picked up the work of Benacerraf to further discuss its implications, for example Resnik[12] and Hellman.[13]

Structuralism has since then developed into many sorts, and there have been many philosophers who write on this subject. The position that will be supposed for this thesis is, as Shapiro calls it, *ante-rem structuralism*, also called *non-eliminative structuralism*, according to which there exist abstract structures: Structures that can be exemplified by any mathematical system of the right kind.[14] Eliminative structuralism tries to *eliminate structures* by explaining them away, such that there do not exist structures in the abstract sense. If there exist any structures, they exist because there exists some system that exemplifies them.

There is quite some recent literature on non-eliminative structuralism. For example, Korbmacher and Schiemer have written on structural properties, giving a formal definition of which properties of a mathematical object can be said to be *structural.*[15] Moreover, Leitgeb has written an article on how non-eliminative structuralism can be used in a concrete field of mathematics, graph theory.[16] Shapiro has written a book on structuralism and the ontology of mathematical objects,[17] and Giovannini and Schiemer have given an (informal) account of how predicates that are defined to be true of a structure acquire their meaning.[18]

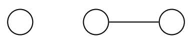

Figure 1: A simple unlabelled (abstract) graph-structure.

Shapiro distinguishes two branches of mathematics. We will restrict us to what Shapiro calls *nonalgebraic mathematics*. The branch of nonalgebraic mathematics studies mathematical fields that are about a single structure up to isomorphism.[19] A good example of a nonalgebraic field of mathematics is the study of the natural numbers: any system of the natural numbers eventually has the same structure. It exemplifies the abstract structure that dictates what relations there must be between the different elements of the system. Another example of a nonalgebraic field of mathematics is graph theory. An abstract graph of the form of Figure 1 can have many instantiations (with different labels), but the underlying structure remains the same. Algebraic fields of mathematics, however, do not just study a single structure. For example, group theory is about many structures that are related, but not the same.[20]

Structuralism gives us a good reason to consider second-order logic as a way of formalization of mathematics. Second-order logic can *categorically* characterize mathematical structures. That is, it can give a characterization such that it is true for all (and only) the

9. Benacerraf, 'What Numbers Could not Be'.
10. Benacerraf, 70.
11. Shapiro, *Philosophy of Mathematics.*
12. Resnik, 'Mathematics as a Science of Patterns'.
13. Hellman, *Mathematics without Numbers.*
14. Shapiro, *Philosophy of Mathematics*, 85.
15. Korbmacher and Schiemer, 'What Are Structural Properties?'
16. Leitgeb, 'On Non-Eliminative Structuralism. Unlabeled Graphs as a Case Study, Part A'.
17. Shapiro, *Philosophy of Mathematics.*
18. Giovannini and Schiemer, 'What are Implicit Definitions?'
19. Shapiro, *Philosophy of Mathematics*, 41.
20. Shapiro, 40.

systems that exemplify the right structure. In terms of numbers, second-order logic can give a characterization such that *any* implementation of the natural number structure fits the logical description, and no other things do. So, if we want to work with abstract structures and formalize mathematical reasoning about them, second-order logic is a natural choice. Of course there is much more to say about this, which will be done in the thesis.

Most structuralists endorse the view that mathematical concepts are implicitly defined by the theories that use them. For example, the concept of a *natural number* is defined *implicitly* by the Peano axioms. Rather than giving an explicit definition for the meaning of that concept by stating what objects it denotes, an implicit or structural definition postulates the requirements for an object to be denoted by the concept. From a structuralist perspective, these requirements might be relational and somehow uniquely determine what it means to be a natural number. Giovaninni and Schiemer have extensively studied implicit and structural definitions.[21] However, their account is largely informal. Moreover, it is still unclear what the actual meaning of such a formula amounts to. We might know *how* the meaning is given to a formula, but has not yet been given an account of *what* the meaning is.

We will argue that the meaning of mathematical statements in a structural context, hence of second-order logical formulae, is an *isomorphism class* of mathematical structures that satisfy the propositional function of the formula. The meaning of a formula thus consists of all structures that are structurally equivalent to a satisfying structure. We will develop this view by giving the context of second-order logic and structuralism and studying the theory of implicit definition. We will then see that this aligns with the above claim.

The thesis is divided into two parts. In the first part, we will lay out the basics of formal and logical languages, and we will discuss the formalization of mathematics in logic. The first thing we will do, is giving an account of formality. Logic makes use of formal languages, so it is necessary to make clear what such a formal language is. Making use of Dutilh-Novaes's different conceptions of formality as written in her article,[22] we argue that *formality as indifference to particulars* is what underlies the step of formalizing statements about (abstract) structures.

Then, we will look at Shapiro's account for second-order logic. There are certain properties that characterize second-order logic, completeness and categoricity, that heavily influence the choice for second-order logic as a way to codify mathematics. We will discuss these properties and discuss the most relevant arguments to choose for second-order logic if we want to codify mathematics as structuralists. We will also discuss the first-order and second-order theories of arithmetic, so we can concretely see their differences. The important notion of non-standard models, related to categoricity, will also be covered.

In the second part of the thesis we will move towards the semantics of second-order logic. We will lay out the differences between proof-theoretic and truth-conditional semantics and will argue in favour of the latter, relating it to incompleteness and to our position of realism. We will then discuss the three different semantics that exist for second-order logic: Henkin semantics, first-order semantics and full second-order semantics. It will be clear that only full second-order semantics will do for our goals.

After having made clear how second-order semantics works, we can give an account of what are called *structural definitions*: Predicates and formulae in structural contexts acquire their meaning in a typical way, that is informally given by Giovannini and Schiemer. We will formalize their theory of implicit and structural definitions. Then, making use of the concept of structural definition, we will give an account of the meaning of a second-order formula in structural contexts, thus answering the research question.

21. Giovannini and Schiemer, 'What are Implicit Definitions?'
22. Dutilh Novaes, 'The Different Ways in which Logic is (said to be) Formal'.

Part I

Formalization and Mathematics

The goal of this thesis is to find out what the meaning of second-order formulae is in a structuralist mathematical context. In this part, we will give the background information that is necessary to understand the second part of the thesis, in which we will answer the research question. As second-order formulae are part of a formal language, we will first give a short introduction in formal languages and, more precisely, *second-order* formal languages. After that, we will discuss two notions of formality, introduced by Catarina Dutilh-Novaes, that are relevant for our goal of finding out the meaning of mathematical second-order formulae.

When we have discussed the different notions of formality, we will have a closer look at the relation between second-order logic, its properties and mathematics. Moreover, we will see why Stewart Shapiro argues for second-order logic as the way in which we can understand mathematics best, relative to its characteristics.

1 Formal Second-order Languages

A formal language, as opposed to a natural language, is an artificial language made for special purposes. Formal languages are, in formal terms, an arbitrary set of words w over some alphabet Σ.[23] More concretely we find a definition in Sider (2009):

> Why are formal languages called "formal"? (They're also sometimes called "artificial" languages.) Because their properties are mathematically stipulated, rather than being pre-existent in flesh-and-blood linguistic populations. [...] Further, formal languages often contain abstractions, like the sentence letters $P, Q, \ldots$ of propositional logic. A given formal language is designed to represent the logical behavior of a select few natural language words; when we use it we abstract away from all other features of natural language sentences.[24]

Formal languages are thus formal by virtue of their properties (such as grammar) being stipulated by us, and by being more abstract than natural languages; we abstract away from natural language artefacts that do not add to the *logical behaviour* of the language.

An example is the logical formalization of the sentence 'John is tall and John is a human.' Since we are only interested in the *logical* form of the sentence, we abstract away from its meaning, except for the logical connective 'and'. So, we get $p \wedge q$, ignoring the actual content of the sentences represented by p and q.[25] So, in short, a formal language is an abstract, artificial language that represents certain characteristics of natural languages. We can then use rules (and in some cases even rule-based machines) to determine whether a sentence is a member of the language.[26] Strictly speaking, formal languages do not have a meaning. Although some of the symbols do represent certain constructs of natural languages, the meaning of the language is decided by us in terms of a *formal semantics*.[27] We will, however, see that intuitively, there still remains some meaning in these formal languages.

23. Sipser, *Introduction to the Theory of Computation*, 14.
24. Sider, *Logic for Philosophy*, 4.
25. Sider, 4.
26. Sipser, *Introduction to the Theory of Computation*, 14.
27. Shapiro and Kouri Kissel, 'Classical Logic'.

Having seen the definition, we can discuss what formal languages are useful for: they can take away ambiguity and lack of clarity that inevitably come with natural languages. Think of a programming language as a means of concisely instructing a computer what to do: giving these instructions in a natural language like English would be a tedious job. Likewise, in logic there is also a great use for formal languages: Using multiple negations in a sentence in English makes that sentence almost incomprehensible. A formal language can make such a sentence easier to understand.

Formal languages *in logic* can thus be used to make clear how certain inferences work; natural language sentences can be translated to a formal language. It is not only possible to translate natural language expressions to formal sentences (think of: the cat lies under the table); we can as well use formal languages to formalize mathematical statements that are expressed in *mathemateze*, the language used by mathematicians. Think of: The number three is larger than the number two.

The standard example type of formal languages is the set of *first-order* languages. These are such that we can express properties of objects. We can say that an object x has a property P. In addition to this, first-order languages allow us to *quantify* over objects. Objects cannot only be denoted by variables (a variable x can be individuated by any object that has the right properties), we can also state that *there exists* an object such that it has property P, or that *all objects* have property P. So, for instance, the sentence *every ball is red* can be expressed by $\forall x B(x) \to R(x)$ with B standing for 'is a ball', R standing for 'is red' and $\forall$ being read as 'for all'. Formally, we have a language with as our alphabet Σ logical symbols $\neg, \to, \forall$, parentheses, variables $x, y, \ldots$, predicates $F, G, \ldots$ and object names $a, b, \ldots$ such that:

(i) if Π is an n-ary predicate and $\alpha_1, \ldots, \alpha_n$ are terms (variables or object names), then $\Pi(\alpha_1 \ldots \alpha_n)$ is a formula

(ii) if φ and ψ are formulae and α a variable, then $\neg\varphi, (\varphi \to \psi), (\varphi \leftrightarrow \psi), (\varphi \vee \psi), (\varphi \wedge \psi), \exists\alpha\varphi$ and $\forall\alpha\varphi$ are formulae

(iii) Only strings that can be shown to be formulae according to (i) and (ii) are formulae.[28][29]

This results in a language in which we can formally express sentences like „all white swans are non-Australian" by having a predicate S for being a swan, A for being Australian and W for being white.

$$\forall x((S(x) \wedge W(x)) \to \neg A(x)) \tag{1}$$

For all things that are swans and white, it follows that they are not Australian. In formula 1 we see that we have abstracted from the meaning of the actual words (like 'swan') and have replaced them with a variable and predicates. Only the *logical behaviour* as mentioned by Sider is kept.

We can generalize our intended natural language even further by choosing a *second-order* language. In second-order languages, we cannot only denote objects by variables, but also properties. Moreover, we can also quantify over properties. The recursive definition given above for first-order languages, may now also be used for second-order languages, with the addition of *property and relation variables* $X, Y, \ldots$ such that, in (i), Π can be either a predicate (property) or a property or relation variable. Moreover, in (ii) also $\forall X\varphi$ and $\exists X\varphi$ are formulae.[30]

28. Sider, *Logic for Philosophy*, 116.

29. For more details or explanation of the recursive definition given above, please refer to Sider or Boolos, Burgess and Jeffrey, *Computability and Logic*

30. Väänänen, 'Second-order and Higher-order Logic', 2.

For example, Russell used the sentence „Napoleon had all the qualities of a great general" as an example of a sentence that cannot be expressed using first-order logic.[31] This means that *for all* properties such that these properties belong to a great general, Napoleon has them. Formally, this would be

$$\forall X(\forall x(G(x) \to X(x)) \to X(n)) \tag{2}$$

with X standing for some property, x for some object, G for being a great general and n for Napoleon.

Another we find in the identity of indiscernibles: If two objects share all their properties, then they must intuitively be identical. This can be expressed by the following formula:

$$(\forall X(X(x) \leftrightarrow X(y)) \to x = y) \tag{3}$$

The formula states that any two objects that have all properties in common, actually are identical. This could of course be challenged, but it goes beyond the scope of this thesis to discuss that here.

We can express certain natural language expressions very easily in second-order formal languages. In the case of the identity of indiscernibles, it is rather clear what the intended meaning of such a formal expression is. If, however, the expressions become more and more abstract, for example when describing a structure, the meaning of such a formal sentence can be less clear. Moreover, we have not yet seen how we can *formally* express the meaning of, for example, Formula 3.

Strictly speaking, as we have seen in the definition of a formal language, the formulae as mentioned above do not have a meaning; they are abstractions from natural language sentences and the meaning should be given by our semantics. However, intuitively, we would say that there *is* some kind of meaning attached to the formula. For example, in formula 3 we simply see that there is some relation between x and y: they both have the properties the other has. That relation, on its turn, is given by $\leftrightarrow$.

When looking at the connective $\leftrightarrow$, we would say that its meaning is preserved while abstracting away from the meaning of the individual objects for which x and y may stand. If the meaning of $\leftrightarrow$ is preserved, then our intuition that $\forall X(X(x) \leftrightarrow X(y))$ has a meaning also makes sense. But what *is* this meaning in a mathematical context? Or, even stronger: Is there still a kind of meaning preserved after abstracting away from the meaning of $\leftrightarrow$?

In the next chapter, we will see what formality actually means: what is it for a language to be formal and in what different ways can we formalize a language? After we have looked into this question, we will see how Shapiro uses second-order logic for his project of codifying mathematics. Besides this, we will also have a look at the different properties that characterize first- and second-order languages.

In the second part of the thesis, we will see the different types of semantics that can be used to give meaning to a formal language. Moreover, we will see the two best-known types of model-theoretic semantics that are being used to give meaning to second-order logic. Having discussed these theoretical issues, we will give an account of how the meaning of second-order formulae is given according to the structuralist thesis. Lastly, we will give a structuralist account of the actual meaning of second-order formulae in such structural contexts.

31. Russell, *My Philosophical Development*, 93.

2 Formal Languages: Abstraction and Meaning

Since we want to find the meaning of mathematical second-order formulae as exemplified before, we will study meaning in a *formal language*, as second-order languages are formal. In order to understand the meaning of a formal sentence, it is useful to understand what we are about to analyse: We need to make clear what our *analysandum* is. What is the reason for mathematics to prefer a formal language over ordinary language and what does the process of formalization do with the meaning of a sentence? In this chapter, I will discuss the article *The Different Ways in which Logic is (said to be) Formal* by Catarina Dutilh Novaes. We will see the different types of formality Dutilh-Novaes distinguishes and we will see how the process of abstraction affects the meaning of a sentence.

As we have seen in the previous section, languages as we use them in logic are *formal*. By formalizing a language one can abstract from different things. The result always is a language with certain characteristics, but what is abstracted from (and so, what stays individuated [32]) can differ.

In her article *The Different Ways in which Logic is (said to be) Formal*, Dutilh Novaes distinguishes eight different notions of formality in logical languages. There is one main difference between more subtle variations of formality in logical languages: one could see formality as pertaining to forms, or as pertaining to rules.[33] I will discuss these two clusters, after which I will discuss two notions that are most useful to us.

Formality as pertaining to forms can be seen as a an Aristotelean notion coming from the distinction between form and matter.[34] Individuated objects are both matter and form; their form is responsible for their essence, what they *are*. Their matter is responsible for what they are *made of*. In a more mathematical context, we can see individuated structures as being the actual objects, made of atomic objects that are individuated on its turn. The form then is the *abstract structure*; no actual objects are used in a formal structure. In a sense, the abstract structure does not have any *matter*. Formality in this sense, thus allows for generality: any object that has the right 'form' can be seen as being an individuated copy of an abstract structure.

Formality as pertaining to rules has a different background. It is more focused on actions than on 'essence of being'. This cluster of notions of formality is more concerned with what logicians and mathematicians *do* and tries to formulate laws that either *describe* what happens or even are the foundation of what happens.[35]

We will argue that the formal as pertaining to forms is the most interesting cluster of formality in our case, given that the use of (formal) logic in mathematical contexts is more focused on being able to generalize, than on being able to do mathematics by strict rule-following.

Generality in structuralism would mean generalizing *towards the form* of the subject that we study.[36] Abstracting away from 'matter' then is: abstracting away from individuated structures and trying to get to the core of mathematics: abstract structures; the form of the structures we are studying. Rule-following here means: being able to make rules that govern the *way* of *doing* mathematics. This could be interpreted as being able to *compute* the outcome of our mathematical argument (whatever that would be), or at least being able to give a set of rules that cover our *practice* of doing mathematics.[37] The main difference thus comes down to being interested in what the objects of mathematics *are* or in what

32. With individuated, I mean: not abstracted from, thus referring to an actual object.
33. Dutilh Novaes, 'The Different Ways in which Logic is (said to be) Formal', 304.
34. Dutilh Novaes, 304.
35. Dutilh Novaes, 321.
36. Dutilh Novaes, 306.
37. Dutilh Novaes, 321.

mathematicians *do* when they study mathematical objects. In this thesis, the main question is what a formal (second order) sentence *means*, which is a more fundamental question that could not be answered in a satisfying way by what is done in mathematical practice. Hence, we will discuss the formal as pertaining to forms in more detail.

2.1 Formality with Respect to Forms

The 'formal as pertaining to forms', a cluster of notions of formality that is focused on what the formal language *is* and abstracts from actual object towards a more general structure or form, is, according to Dutilh Novaes, again divided in subcategories: formal as abstraction from subject-matter and formal as variability.[38] Variability is a main reason to choose a formal language in our case: the ability to say something about *any* mathematical object with the right properties is extremely useful in mathematics. For example, when we want to say something about the common structure of 'Ada loves Betty' and 'Betty loves Charles', we could abstract to a formal language such that the sentence $L(a, b) \wedge L(b, c)$ can be used to say things about *any* sentence of the same form. Likewise, we would also like to be able to abstract from a sentence about, say, a graph, so that it applies to any graph of the same structure. A difference with formality as abstracting from subject matter is that we still want the formulae to be applied to a concrete subject eventually.

Abstracting from subject-matter can be read as: generalizing logical languages in such a way that the subject-matter (content) of expressions is taken away: we do not need to know what the individual parts of a sentence stand for, in order to do logic. The example sentence 'Ada loves Betty and Betty loves Charles' illustrates nicely that the content of the sentence is not necessarily interesting to us, we just want to know its logical form. Whether it is about love, Ada, Betty or Charles does not matter; we just study three objects that bear some relation to each other.

Both subcategories are interesting to us. We will especially discuss formality as indifference to particular objects and formality as de-semantification. The former is a subtype of formality as variability and turns out to be of great use for the structuralist project: it allows for generalizing towards more abstract objects, whereas the latter is a subtype of formality as abstraction from subject-matter and is most often used by logicians.[39] According to this position, symbols should be treated itself as mathematical objects, not having a meaning at all.[40] This subcategory also allows for generalization, but abstracts further away from meaning than the formality as indifference to particulars. We will discuss the usability of formality as de-semantification for mathematical goals.

2.1.1 Formality as Indifference to Particulars

The programme of formality as indifference to particulars works, as mentioned earlier, by abstracting from the matter of an object (in our case this is an individuated structure) while keeping the *form* of that structure. [41] By abstracting towards the structure, the form, of a sentence, we can say something about *all* sentences with that structure. By replacing actual

38. Dutilh Novaes, 'The Different Ways in which Logic is (said to be) Formal', 306.
39. Dutilh Novaes, 314.
40. Dutilh Novaes, 314.
41. The programme of indifference to particular objects is closely related to Tarski and Corcoran, 'What Are Logical Notions?' according to Dutilh Novaes, 'The Different Ways in which Logic is (said to be) Formal', 312. In that article, Tarski researches logicality: what notions are logical, meaning what notions are preserved when changing the meaning of all objects? Notions like 'and', 'or' are preserved, even when all objects in the universe are permutated. Logicality then means: abstracting away from particulars in such a way that the actual objects are variable, but logical notions are not.

names standing for objects by terms, such as variables a and b, we can say something about all sentences with this form.

In a structural context, this approach of formality could help us with understanding abstract mathematical structures. The structures we try to study are characterized by logical formulae describing an *abstract structure*, so we really are indifferent to particulars; we want to say something about all mathematical structures of the same form. If this is what the formality is about, the symbols being used still bear certain relations to each other, while we abstract away from the actual 'particulars' towards a more variable, thus formal, description of the structure. We could say that structural meaning is being preserved, also knowing what this structural meaning is; if an abstract sentence ψ follows from an abstract sentence φ, we can say that any two sentences with the same form bear this relation. Still, in our formal approach, the reference to particular objects is absent.

In the sentence $(P(x) \rightarrow Q(x))$, x is a *free variable.* A free variable denotes a particular object, but we do not specify it; we are indifferent to this particular.[42] Likewise, we are indifferent to which objects actually have the properties stated in the formula (but *if* they are P, then they are Q). Any object that does comply with the sentence can be used as instantiation of x. This is just a simple example of using a formal language to abstract from particular objects towards the structure of a sentence. There are of course many more cases in which this notion of formality applies. Think of prime numbers: we can state that *if* some member of a natural number structure is prime and larger than two, *then* it is uneven. What object it applies to, however, is not part of the formula yet.

This is exactly what makes this position interesting to us: when we abstract away from individuated structures, such that we end up with variable abstract structures which are individuated by all structures that share that same form, structural properties are still preserved.[43]

In the next chapter, we will see what the application of this notion of formality is to second-order logic in mathematical contexts, but it is clear that indifference to particulars is a main reason for us to consider formalizing the way we talk about abstract structures.

2.1.2 Formality as De-semantification

The other notion of formality of logical languages we will consider, is the approach of de-semantification. On this view, to be purely formal amounts to manipulation symbols as blueprints with no meaning at all, as pure mathematical objects and thus no longer as signs that stand for something more concrete.[44]

In the process of abstraction, we abstract away not only from the meaning of objects in a sentence, but even from the meaning of the symbols we use to express relations between objects or variables. Dutilh-Novaes quotes Bernays, who gave a specification of the goal of the programme: We abstract away from the contents of a sentence, but even more heavily than we did in the other programmes. We disregard the original meaning of the logical symbols (whereas in the other case, meaning of logical terminology was preserved), and we make the symbols like $\leftrightarrow$ themselves representatives of formal objects and connections.[45]

Abstraction then not only means abstraction from individuated object to the *form* or structure of that object, but also abstracting away from the *meaning of signs.*[46] 'Signs' in this context are *logical symbols*, symbols like $\wedge, \rightarrow$. Abstraction from the meaning of logical

42. Sider, *Logic for Philosophy*, 131.
43. We will come to structural properties later.
44. Dutilh Novaes, 'The Different Ways in which Logic is (said to be) Formal', 318.
45. Dutilh Novaes, 319.
46. Dutilh Novaes, 319.

symbols allows for 'doing logic' by just manipulating symbols, without knowing what they stand for. In the sentence $(P(a) \to P(b)) \vee P(c)$ the symbols have no more meaning than the sentence letters; we can manipulate the symbols by following rules, but we cannot reason with them in a semantic way.

We could however state that, intuitively, there is a small part of the meaning of symbols that is preserved. In the sentence $(P(x) \to Q(x))$, we have the intuition that the symbol '$\to$' is such that it expresses a relation between $P(x)$ and $Q(x)$. We might have abstracted away from the meaning of that relation, but just the fact that there is a relation is still present. We can thus say that a certain *structural* element of the meaning of symbols is preserved in the process of de-semantification.

We have now seen two different notions of formality, the notion of formality as indifference to particular objects and the notion of formality as total de-semantification: detachment from the meaning of even logical symbols. Although the approach of de-semantification is the most recent notion of formality in logic,[47] mathematical contexts require a different approach, which we find in formality as indifference to particulars.

In order to be able to make a choice between the two notions of formality and decide which one suits our programme best, we need to understand the role of second-order logic in mathematical contexts. In the following chapter, we will discuss the programme of Stewart Shapiro, and see in what way second-order logic can be used to express mathematics.

3 Shapiro's Case for Second-order Logic as the Language of Mathematics

In his book *Foundations without Foundationalism*,[48] Stewart Shapiro argues for a fundamental role for second-order logic in mathematics. We will discuss the case Shapiro makes, and we will see what this means for the notion of formality that is needed. After all, we want to know the meaning of second-order formulae, which has both an intuitive and a technical (semantic) side. By discussing the applicable notion of formality, we can get a grasp of the intuitive meaning of a second-order formula in a mathematical context.

First of all, Shapiro gives different ways in which logic can be of use. The main question then is: „What is the best logic (or language) in which to (...)?"[49] The most interesting item to fill in the blank in our case is the case for using logic to formalize a particular branch of mathematics. This can involve both codifying the 'truths' of this branch, or describing the structure studied by that branch.[50] Since we try to find the meaning of a second-order formula that characterizes some structure, we will be concerned with the latter.

Note that using a form of logic as a means to the mentioned end means that we design a formal language to formalize not a natural language like English, but a 'natural mathematical' language, that we sometimes call *mathemateze*. An expression in mathemateze might be 'a graph with two connected nodes' or 'the set containing all and only the even natural numbers'.

First, we will look into an important property of first-order logic: the property of completeness. This property plays an important role in the process of choosing a certain kind of logic. We will then see what Shapiro's conception of logic exactly is, and how that relates to the notions of formality we have discussed in the previous chapter. We will then discuss

47. Dutilh Novaes, 'The Different Ways in which Logic is (said to be) Formal', 318-19.
48. Shapiro, *Foundations without Foundationalism*.
49. Shapiro, 10.
50. Shapiro, 10.

the role logic could play in being a foundation of mathematics, after which we will look into some more technical details concerning second-order logic.

3.1 Completeness and Categoricity

As we have seen in the previous chapter, there are several *orders* in terms of quantification in formal languages. We restrict ourselves to two of those: first- and second-order languages. Whereas first-order languages allow for quantification over objects (in the sense of 'all objects having property P'), second-order languages allow for quantification over *properties* (or predicates): We can express an object having all properties of a certain kind in second-order languages. There are important properties a logical system can have that derive from these differences in expressive power of the two languages. We will discuss two of them in this chapter: the property of completeness and the property of categoricity.

The concept of completeness applies to the relation between a semantics and a deductive system for a logic. When we design a logical language $\mathcal{L}$, we first decide the syntax (grammar) of the language, and then introduce a *semantics* for this language, in which we decide the meaning of different expressions of $\mathcal{L}$. The semantics of a language consists of a formal way of defining what makes an formula a *true* formula and when one formula *entails* another.[51] We do this by defining the notion of a *model* in which we model the situations we want to discuss and in which we define what formulae are true. A model $\mathcal{M}$ then consists of a *domain of discourse* $\mathcal{D}$ and an interpretation function $\cdot^{\mathcal{M}}$, since the notion of truth is given under a certain *interpretation*.

With an interpretation, we mean the way of interpreting the non-logical vocabulary of our language which is not already fixed within the language definition itself (which we call the logical notions, also see Tarski's *What Are Logical Notions?*[52]).[53] If it is true that the interpretation of a set of formulae Γ is such that a formula φ is also true under that interpretation, and this is the case for all models (hence, under all variations of the meaning of non-logical symbols), we write that $\Gamma \vDash \varphi$. We then say that Γ *implies* φ. When a defined *model* is such that a formula φ is true in that model, we write $\mathcal{M} \vDash \varphi$.[54] We then say that $\mathcal{M}$ *makes* φ *true*. An interpretation of a piece of vocabulary σ of a language $\mathcal{L}$ by a model $\mathcal{M}$ is written as σ^M.

The interpretation of a predicate $R^{\mathcal{M}}$ is defined by giving its *extension*: In a model we fix the reference of a predicate. We define the extension of a predicate R by giving the set of objects in the domain with property R. In the case of domain $\{1, 2, 3, 4\}$ the interpretation of predicate E 'is even' is $E^{\mathcal{M}} = \{2, 4\}$.

Besides a semantics, we can also design a *deductive system* which tells us how we can infer logical truths in a syntactic manner. There are systems that allow us to do so by following rules from certain starting points called *axioms*.[55] More generally, a deductive system is a syntactic system that consists of a fixed set of rules such that one formula can be derived from another. A deductive system allows us for making deductive inferences (without having to know what the formulae we use actually mean). If a formula φ is syntactically deducible from a set of formulae Γ, we write $\Gamma \vdash \varphi$.[56]

Naturally, we want that the notion of implication and that of deduction to coincide: if a set of formulae implies some formula, it would feel counter-intuitive if our deductive system did not allow for proving the implication. This is the point at which the notion of soundness

51. Montague, *Formal Philosophy*, 223, footnote.
52. Tarski and Corcoran, 'What Are Logical Notions?'
53. Shapiro, *Foundations without Foundationalism*, 5.
54. Awodey and Reck, 'Completeness and Categoricity. Part I', 2.
55. Shapiro, *Foundations without Foundationalism*, 4.
56. Awodey and Reck, 'Completeness and Categoricity. Part I', 2.

and completeness comes in: if a logical system is complete, any sentence that is semantically implied by another set of sentences can also be proved by the chosen deductive system. If it is sound, any provable inference is also a semantically valid argument.

In their paper on completeness and categoricity, [57] Steve Awodey and Erich H. Reck give a very comprehensive account of what completeness and categoricity are and their history. Given a syntax and a semantics for a language $\mathcal{L}$, we define the following.

Definition 1 (Completeness) *The deductive consequence relation* $\vdash$ *is called* complete *relative to the semantic consequence relation* $\vDash$ *iff for all sentences* φ *and all sets of sentences* Γ *of* $\mathcal{L}$ *it holds that if* $\Gamma \vDash \varphi$*, then* $\Gamma \vdash \varphi$*.*[58]

Informally put, if a logical system has the completeness property, any argument that we can make semantically, also holds syntactically. So, if we can infer that 'all humans are mortal, Socrates is a human, therefore Socrates is mortal' is a semantically valid inference, in a complete system this also means that we can infer this sentence by means of derivation. This works the other way around when talking about soundness: in a sound system, if we can prove that 'Socrates is mortal' follows from premises 'all humans are mortal, Socrates is a human', this is also a semantically valid argument. We therefore define soundness as follows:

Definition 2 (Soundness) *The deductive consequence relation* $\vdash$ *is called* sound *relative to the semantic consequence relation* $\vDash$ *iff for all sentences* φ *and all sets of sentences* Γ *of* $\mathcal{L}$ *it holds that if* $\Gamma \vdash \varphi$*, then* $\Gamma \vDash \varphi$*.*[59]

Because of the importance of the concept for codifying mathematics in logic, we will define categoricity: when a logic is *categorical* this means that there exists a theory that can be interpreted by the logical, such that the theory $\mathbb{T}$ has *essentially* only one model. To give a formal account of this notion, we first need to discuss the concept of *isomorphism*. Intuitively, two objects are *isomorphic* when they share the same 'form': they are structurally the same. This means that two isomorphic objects are structurally indiscernible.[60] In terms of logical languages, two *models* can be isomorphic which means that they share the same structure. This also yields the result that the two isomorphic models have the same set of truths.[61] An isomorphism between two structures is defined as follows.

Definition 3 (Isomorphism in a logical system) *If* M *and* N *are models of a language* $\mathcal{L}$*, then* M *is isomorphic to* N *iff there is an bijective function*[62] f *which assigns to each member of the domain of* M *a member of the domain of* N*, such that* f *applied to an interpretation of a member* σ *of the non-logical vocabulary* $\mathcal{L}$ *by* M *yields the interpretation of* σ *by* N*:* $f(\sigma^M) = \sigma^N$*.*[63]

A function f as above preserves the structure of M: if it exists then everything M models can also be modelled by N and since each interpretation of the language by M is *uniquely* mapped to an interpretation of the language by N, we can say that all differences between the two models are purely related to content: from a structural point of view, the models are exactly the same. We write $M \simeq N$.

57. We will discuss the notion of categoricity later in this chapter.
58. Awodey and Reck, 'Completeness and Categoricity. Part I', 2.
59. Halbeisen and Krapf, *Gödel's Theorems and Zermelo's Axioms*, 40.
60. Corcoran, 'Categoricity', 190.
61. Corcoran, 190.
62. We say a function is *bijective* when it assigns to each member of the domain a *unique* member of the co-domain and when each member of the codomain is assigned to a member of the domain.
63. Corcoran, 'Categoricity', 196.

Now that we know what an isomorphism between two models is, we can look into the notion of categoricity. We say that a theory, a set of axioms of some logical language $\mathcal{L}$, is categorical if it essentially has only one model. That is, all models we can think of that make the axioms true, boil down to being isomorphic. In more formal terms, we define:

Definition 4 (Categoricity of a theory) *A theory* $\mathbb{T}$ *is* categorical *relative to a semantics, iff for all models* M, N *of* $\mathbb{T}$ *in that semantics,* $M \simeq N$*.*[64]

There are some well-known examples of categorical second-order theories, given a certain semantics: one is second-order Peano arithmetic, which categorically expresses our 'standard' arithmetic. Hence, all second-order models of Peano arithmetic are isomorphic. We will discuss this theory later, but for a very short introduction of second-order Peano arithmetic, one can refer to Awodey and Reck.[65] First, we will discuss concisely in what logical systems the completeness property holds and what its importance is.

Without going into completeness and incompleteness in too much detail, we will discuss the results of the completeness theorem and the incompleteness theorem of Kurt Gödel. According to Gödels completeness theorem, which we find in his 1930 paper,[66] the completeness property holds for first-order languages with a deductive system equivalent to the deductive system to which Gödel's proof initially applied, and a complete semantics (that is, a semantics that renders any formula either true or false). In short, Gödel's completeness theorem states that Definition 1 holds for first-order logic: there exists a deductive system such that every semantically valid formula is derivable.[67]

However, for standard second-order logic, there cannot be such a completeness proof. Gödel proved that any theory which implements arithmetic is inherently *incomplete*. His proof is about a different kind of completeness, *negation completeness*, but from that one can infer that second-order logic lacks the type of completeness we are interested in. Negation completeness is defined as follows:

Definition 5 (Negation completeness) *A theory* $\mathbb{T}$ *is called* negation complete*, iff for all sentences* φ*, either* $\mathbb{T} \vdash \varphi$ *or* $\mathbb{T} \vdash \neg\varphi$*.*[68]

Definition 6 (Semantic completeness) *A theory* $\mathbb{T}$ *is* semantically complete *iff for all sentences* φ*, either* $\mathbb{T} \vDash \varphi$ *or* $\mathbb{T} \vDash \neg\varphi$*.*[69]

Gödel showed in his article that any ω-consistent theory that can interpret first-order Peano arithmetic cannot be *negation complete*. There are sentences such that neither the sentence nor its negation can be proved from the axioms in $\mathbb{T}$.[70]

As we will see, second-order logic can be used to *categorically* (see Definition 4) axiomatize second-order Peano arithmetic.[71] Since second-order Peano arithmetic is also an implementation of arithmetic, second-order Peano arithmetic cannot be negation complete either. Hence, second-order logic can never be negation complete as it is possible to interpret Peano arithmetic in second-order logic.

64. Awodey and Reck, 'Completeness and Categoricity. Part I', 3.
65. Awodey and Reck, 7.
66. Gödel, 'Die Vollständigkeit der Axiome des logischen Funktionenkalküls'.
67. Gödel, 350.
68. Awodey and Reck, 'Completeness and Categoricity. Part I', 4.
69. Awodey and Reck, 3.
70. Gödel, 'Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I', 174.
71. Awodey and Reck, 'Completeness and Categoricity. Part I', 7.

Any theory that is categorical is also semantically complete.[72] It is a mathematical fact that, since second-order Peano arithmetic can be categorically characterized in second-order logic, the implementation of second-order Peano arithmetic must be semantically complete as in Definition 6.[73] But from Gödels theorem it follows that the theory $\mathbb{T}$ of Peano arithmetic is not negation complete as in Definition 5. Hence, there will always be some semantic facts that cannot be deduced in any deductive system. That means that the language we are working in does not comply with Definition 1. Because of its categoricity, second-order logic is incomplete.[74]

We will now discuss what influence the property of completeness and that of categoricity has on the choice of a logical system (that is, the choice between first- or second-order logic). In order to do so, we will first discuss the debate on, as Shapiro calls it, *foundationalism* in logic.

3.2 Conceptions of Logic

Shapiro mentions two conceptions of logic. We will show that there are some parallels between those conceptions and Dutilh-Novaes notions of formality as discussed in Section 2.

Shapiro discusses the notion of the *foundationalist conception of logic*, as opposed to the *semantic conception of logic*. According to the foundationalist conception of logic, an absolutely secure (or as secure as humanly possible) logical foundation of mathematics exists.[75] According to Shapiro, foundationalists are usually interested in the deductive system of a logic.[76] Shapiro does not give further arguments to support this claim, but there are some good reasons to think he is right, given his interpretation of this foundationalist conception of logic. I will give the argument for axiomatic proof systems, but it goes analogous for other deductive systems as well.

For a foundationalist, it is important that inferences are absolutely certain in order to provide a solid foundation for mathematics. Moreover, the reasoning that is being used in such inferences should be *self-evident*; there should be no reasonable way to doubt the rules used in the inferences.[77]

Especially the fact that deductive systems allow for more clarity on their way of inferring truths seems like a good reason for an advocate of the foundationalist conception of logic to prefer a reasoning in a deductive system over semantic reasoning. Codifying true propositions in a branch of mathematics can of course in principle be done by defining a semantics, but proving true statements in such a system is less clear than syntactic proofs.

When we want to prove that a formula φ follows from a set of formulae Γ, I argue that proving this deductively provides more clarity than the semantic proof for the same proposition. With clarity, I here mean that a proof can be followed by any subject that has basic reasoning skills. First of all, the starting axioms are clear. Then, when a syntactic argument is given, this is done on a step-by-step basis with a fixed set of rules. In order to give a correct proof in a deductive system, every step needs to be made clear and should be written down; inferences can only be made with the rules provided beforehand. One does not have to know anything about the justification of the rules in order to see that they are applied correctly: whether the rules are applied correctly can be seen very quickly just because of the step-by-step reasoning that is given. If φ can be deduced from the axioms and the formulae in Γ by the rules given, one does not need to know the meaning of either φ

72. Awodey and Reck, 'Completeness and Categoricity, Part II', 83.
73. Awodey and Reck, 'Completeness and Categoricity. Part I', 4.
74. Shapiro, *Foundations without Foundationalism*, 8.
75. Shapiro, 35.
76. Shapiro, 36.
77. Shapiro, 35.

or the formulae in Γ, the axioms or the rules: seeing that a rule is correctly applied requires only basic capacities in reasoning.

This property does not hold for semantic proofs: usually a proof that a formula φ semantically follows from a set of formulae Γ requires more words and text than a deductive proof. This is not a big problem, but moreover, a semantic proof requires a basic conception of the meaning of the formulae; by assuming the antecedent formulae in Γ one can, by the meaning of these formulae, argue that φ is also true.

The correctness of such a proof is not as clear at one glance as is the case with a deductive proof: Instead of a fixed set of rules, *any* rule can be used, as long as it is justified correctly. These rules appeal to the meaning of the formulae used, and in order to see whether they are justified or not, one needs more than just a basic *intuition* of what is valid reasoning and what not. Even if the semantic proof is given step-by-step, a reader would still need a grasp of the meaning of the formulae but it also takes quite some reasoning to get one's head around the different steps taken in the proof. As there is no fixed set of rules, *everything goes* as long as the rule can be justified. That demands a lot more of the reader of the proof, so for the need of self-evident reasoning, syntactic proofs are the better choice: one can directly see the validity of such a proof, as opposed to semantic proofs.

Besides this reason of clarity, there is another thing which can be said in favour of deductive systems if one wants to give a foundation of mathematics: there can be multiple interpretations of a logical system of which all valid inferences can be codified by using a single deductive system. One deductive system can codify many different semantics. This is also something which can be a reason to choose reasoning by deduction over semantic reasoning: if a certain language (say, that of mathematics) is to be formalized, this formal language might be interpreted by several different relevant semantics. If a deductive system can be given such that it is sound and complete with respect to these different semantics, checking validity of an inference in the deductive system requires less work than checking the validity in all relevant semantics.

All in all, we see that, in order to give a foundation of mathematics, focusing on a deductive system has clear advantages. We will assume that if the foundationalist conception of logic is being endorsed, its users are interested in deductive proofs rather than semantics.

The ultimate goal of the foundationalist conception of logic is to find a logical system that can be the *foundation* of a mathematical theory: every ideally justified statement (that is, intuitively valid in the natural language) must also be provable. This foundation should be purely deductive; all possible true inferences according to the mathematical theory should be deducible. Most authors use a deductive system such that this comes down to being deducible from axioms, using the rules defined in the deductive system.[78]

The semantic conception of logic is different in the way that, according to this position, correct inferences are defined in terms of semantic validity. The ultimate goal is not to give a foundation for a certain theory such as arithmetic, but to *model* this theory such that every sentence that is intuitively valid in the 'natural language' of the subject matter, also is a semantically valid inference in the logical system that is being used.[79]

There have been several efforts over the last century, such as the logicist programme, which tried to give a foundation of all mathematics in a logical language.[80] For a somewhat modest attempt we can refer to Nicolas Bourbaki, a group of mathematical philosophers who, in their paper *Foundations of Mathematics for the Working Mathematician*[81] tried to

78. Shapiro, *Foundations without Foundationalism*, 35.
79. Shapiro, 39.
80. Shapiro, 29.
81. Bourbaki, 'Foundations of Mathematics for the Working Mathematician'.

give an account of a logical and set-theoretical foundation of mathematics, from a descriptive perspective.

According to Bourbaki, logic has little purpose outside the context of mathematics.[82] Instead of being useful outside mathematics, we *can* use it for describing what mathematicians do. The system that comes out of this process has no *prescriptive* use, it is based on mathematical texts that are already there and should model the statements made there in an abstract manner, but it should not go beyond that use. The logician should be guided by what mathematicians do (or more precise: have done) and not by what they exactly study. If a logical system is made to be of normative value, it needs to at least allow for the things mathematicians want and not 'try to make him comform to some elaborate and useless ritual'.[83]

Bourbaki points out that such logical analyses of mathematics can help to find contradictions, after which mathematicians refine their theory. In the rest of this article, Bourbaki sets out a formal system that can be used to build on their mathematics up to that date.[84] It basically is an implementation of (weakened) set-theory in first-order logic.[85] It is, thus, an attempt to give a foundation of mathematics in first-order logic, which is in line with the ideas of the foundational conception of logic.

As their system is built in first-order logic, the inferences that can be made are provable due to completeness as in Definition 1. From a foundational perspective this is something important: if we want to give a description of the axiom on which all mathematics can be built (by means of deduction), we must be sure that all true statements in fact can be deduced, otherwise the foundation cannot be used for the ultimate goal of giving the foundations of all mathematics.

Although advocates of the fundationalist conception of logic argued to stick with first-order logic because of its completeness, there are also reasons not to be willing to accept this limitation. It might, for example, be the case that the expressive power of first-order logic is too small. It then is a possibility to introduce the incomplete but categorical second-order logic: Its expressive power is much larger and that might be enough reason to have a closer look at it, even despite its drawback of the lack of completeness.[86]

The foundational conception of logic as found in the work of Bourbaki is related very closely to notion of logic as describing, or formalizing, the *laws of thought*: it formalizes the ways in which rational agents should reason.[87] Logic can, according to this view, be used to find out according to which patterns or laws mathematicians work. This logical programme makes use of broadly the same notion of logical formality as Dutilh-Novaes' notion of *formality as pertaining to rules.*

As we have already seen in the previous chapter, this notion of formality is not as useful for our goals of understanding what mathematical structures *are*, so if we look at the notion of formality used in the *foundationalist conception* of logic, it is not really useful to us.

This is in line with the conclusion that Shapiro draws; he states that it is almost impossible to know or recognize that a certain *deductive system* (the system of axioms and rules) is sufficient: in order to be so, any justified argument in our natural language (mathemateze) would need to have a deduction in the logical formal language. To know or see whether this is the case, is to expect *a lot* from our powers.[88] It would require some kind of 'map of logical space' in order to determine which arguments are deducible; that is something

82. Bourbaki, 'Foundations of Mathematics for the Working Mathematician', 2.
83. Bourbaki, 2.
84. Bourbaki, 3.
85. Rosser, 'Review of Foundations of Mathematics for the Working Mathematician', 248.
86. Shapiro, *Foundations without Foundationalism*, 35.
87. Shapiro, 36.
88. Shapiro, 37.

we simply do not have.[89] The foundationalist conception of logic expects too much from its users to be a feasible option for our programme. Luckily, there is still another conception of logic.

The other conception of logic mentioned by Shapiro is the semantic conception. This conception makes use of a formal language in a way that we called *formality as pertaining to forms* earlier. Although Shapiro does not give a detailed description of the notion of formality preferred by him, we can benefit from knowing what his 'semantic conception' of logic exactly means, in order to see what notion of formality distinguished in the previous chapter fits our goals of understanding mathematical structures best.

3.2.1 The Semantic Conception of Logic

Whereas the *foundational conception* of logic focuses on a deductive system, the *semantic conception* of logic focuses on correct inferences in terms of semantic validity; validity in terms of meaning.[90]

What we ultimately try to find, is a logical system such that an inference is valid, if and only if it is a justified argument in the corresponding natural language: Whether a semantics is plausible, depends on whether the valid inferences correspond to judgements about the natural-language inferences and vice-versa. If the two do not match well, the semantics can be adjusted, or maybe the judgements about natural language inferences.[91] Forming a useful semantics thus is a quite holistic enterprise, according to Shapiro, in which we adjust our intuitions and our semantics step by step, as soon as we find incongruence. A correct model (at a certain point in time) thus forms some kind of equilibrium.

It is the fact that this conception of logic uses semantic inference, rather than deduction, that makes categoricity an important notion here. Categoricity, after all, is a semantic notion itself: it tells us something about all models that exist for a certain theory. If we can *categorically* define a theory, that means we defined it in such a way that there is essentially only one interpretation of that theory. Hence, if a theory defines a structure, it will uniquely do so: All interpretations will be isomorphic. Concretely, if the natural numbers are defined by a categorical theory, there is essentially only one model. The theory thus *uniquely* defines the structure; there are no competing interpretations.

This is the point at which things start to get interesting in terms of structuralism: if two models are the same up to isomorphism, then their structural properties are identical.[92] If we can categorically define a certain mathematical structure using a second-order formula, we have basically defined *all* structures of that 'kind'. Think of a labelled graph: if we manage to describe that structure categorically, the resulting model will describe any graph that shares it structure (that is, the same graph with different labels). The fact that second-order logic can thus be used to describe structures categorically, is of great use for the structuralist programme.

Of course, not *every* theory or field in mathematics can be categorically characterized.[93] There are theories that have many structures that satisfy them. Topology and group theory are such theories: They are about a *class* of structures. Hence, for group theory there is not one unique structure up to isomorphism that satisfies it. As mentioned in the introduction, we will not say much about these theories, that are part of the class of *algebraic fields of mathematics*. It is, however, good to know that categoricity is not something that applies, or should apply, to any theory in mathematics.

89. Shapiro, *Foundations without Foundationalism*, 37.
90. Shapiro, 38.
91. Shapiro, 39.
92. Korbmacher and Schiemer, 'What Are Structural Properties?', 304.
93. Shapiro, *Philosophy of Mathematics*, 40.

Now we can look back at the notion of formality that is tacitly being used by Shapiro. The question comes down to the role the formal language plays in the semantic conception of logic. Sure enough it is not about capturing the rules of mathematical practice in a formal language or give rule-based deductions in order to describe mathematics: finding an equilibrium in the way one has to do in order to model a natural language or phenomenon is not subject to strict rules: there are many ways of doing so.

The notion of formality which is used by this conception of logic must then be one of the cluster that we called *formal as pertaining to forms* as in the previous chapter. That means that either the notion of formality as variability or formality as de-semantification is used. Since Shapiro considers the distinction between logical vocabulary and non-logical vocabulary important, as we would not consider interpreting $(P \wedge Q)$ as 'P or Q' valid, there is some meaning inside the formulae before interpreting them: call this the *intended reading.*[94] Denying this intended reading comes down to playing an entirely different game: such an interpretation of the language does not allow for a serious discussion.

Formalizing the language in such a way that even the symbols become meaningless would result in a language with no logical vocabulary; we can interpret everything the way we want. Even though this might be useful in a deductive system, in which the meaning of symbols is completely irrelevant, this goes too far for the semantic conception of logic. In trying to find a fitting semantics, at least *some things* have to keep their meaning; otherwise only inferences that already have their conclusion as a premise are valid.[95] In semantics, we need the meaning of the formulae to reason about them and to make rules such that we can reason. Since deductive systems work with a fixed set of rules, we can forget about the meaning of sentences once these rules have been fixed.

However, as having hinted at before, the notion of formality as de-semantification *might* keep some of the meaning of logical symbols, even if we treat them as 'meaningless signs'. To keep some kind of meaning, we need to interpret formulae as structures themselves. This can best be seen with an example. The formula $(P(x) \wedge Q(x))$ has a certain underlying structure which makes it rather irrelevant whether $\wedge$, & or P is written, as long as that structure is instantiated. So, if we see the formula $(P(x) \wedge Q(x))$ as having an underlying structure, that structure is instantiated by both $(P(x) \wedge Q(x))$ and $P((x) \& Q(x))$. Structurally speaking, these two formulae are identical; there is no way in which they express different things apart from the symbols. So, even after de-semantification of the individual symbols, there seems to be some meaning left in terms of structure: $P(x)$ and $Q(x)$ have a structural relation, even though it is not clear at plain sight what this relation exactly is.

Although the two conceptions of logic have a different goal, it might well be the case that the foundational and the semantic conception of logic *extensionally* coincide: they amount to the same thing. This is of course so when the logic that is being used is both sound and complete; any inference that is valid in the semantics can also be deduced and any deducible argument can also be proved in a semantic way. Since the foundationalist conception is associated with deductive systems, and the semantic conception is associated with a semantic theory, we can see that the two conceptions ultimately are the same since the deductive system and the semantics allow for deriving the same truths. When the used logic is incomplete, there cannot be such a collapse of the difference between the two conceptions of logic; we then have to choose. Because of the high demands that the foundationalist conception puts on its users and because of the notion of formality that is not really useful for our project, we better go with the semantic conception of logic.

94. Shapiro, *Foundations without Foundationalism*, 39.
95. Shapiro, 39.

3.3 Logic and Codification of Mathematics

Now that we know that the semantic conception of logic fits our goal best, we can look into Shapiro's case for second-order logic to codify mathematics. First, we will reconstruct Shapiro's argument for the use of second-order logic instead of first-order logic, after which we will give a short introduction in some mathematical concepts described by second-order logic. One of those will be the *abstract structure* of arithmetic, in which we are particularly interested.

In order to reconstruct Shapiro's argument for the rejection of first-order logic, it is good to have a running example, which we find in the Peano arithmetic-axioms. The simplest axioms of *first-order* Peano-arithmetic, which only define the natural numbers $\mathbb{N}$, are given by:

(i) $\forall x(0 \neq S(x))$

(ii) $\forall x, y(S(x) = S(y) \rightarrow x = y)$

(iii) $(\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(S(x))) \rightarrow \forall x\varphi(x)$[96]

We use S as representing a successor function (which, in a model, should in fact give the successor of a certain number). Note that the third axiom actually is an axiom *scheme*: it is instantiated for every formula φ that is a member of the first-order language $\mathcal{L}$. So, every instance of (iii) is an axiom. We call this formula the *induction scheme*: it assures us that if something can be said about the first natural number and of every successor of that number, it can be said of *all* natural numbers.

In second-order logic, we replace the induction *scheme* by the induction *axiom* which is second-order:

(iii*): $\forall X((X(0) \wedge \forall x(X(x) \rightarrow X(S(x)) \rightarrow \forall x X(x))$.[97]

Whereas first-order Peano arithmetic consists of axioms (i), (ii) and (iii), we state that second-order Peano arithmetic is axiomatized by (i), (ii) and (iii*). We see that X ranges over the properties of our domain and is not a formula scheme: Peano arithmetic is finitely and categorically axiomatizable in second-order terms.

The most interesting differences between second-order logic and first-order logic (and the shortcomings of the latter) lie in semantical issues.[98] There are some widely used mathematical notions which first-order languages are incapable of expressing. For example, one of these notions is that of finitude and cardinality, the number of elements in a set. Although we can construct a theory such that it only allows for infinite models, we cannot express in a first-order axiomatization of set-theory that a certain set is finite without having to give a maximum number of elements.[99] That is, we cannot state that a set has n elements without having to give the value of n.

Moreover, Boolos has proved that comparisons in terms of cardinality can only been given for specific sets, but not in general terms.[100] The fact that first-order logic is too weak to be able to express these important mathematical concepts show that it might be interesting to accept second-order logic in order to codify more of mathematics.

There is, however, a maybe even bigger problem with first-order theories: they might not be able to cover all of the notions we want them to, but they sometimes cover *too much*. Some first-order theories allow for the existence of *non-standard models*. For a mathematical

96. Shapiro, *Foundations without Foundationalism*, 82,110.
98. Shapiro, 111.
99. Shapiro, 102.
100. Shapiro, 102.

field that has both a first-order and a second-order theory, we define a non-standard model as follows:

Definition 7 (Non-standard model) *A model of a first-order theory is non-standard iff that model is not part of the set of models of the corresponding* second-order *theory.*[101]

The property of being a non-standard first-order model thus boils down to not being part of the models of the respective second-order theory. As the second-order theory of arithmetic is categorical, all of its models are isomorphic to one another. We can therefore sharpen the definition a bit more using this idea:
A model of a first-order theory is non-standard iff it is not isomorphic to the categorical models of the corresponding second-order theory.
We say that, since the models of second-order Peano arithmetic exactly cover the natural numbers (or an isomorphic structure), these models are the *intended models*: they interpret arithmetic in the way we meant it. Of course this does not automatically follow for any theory (there might be first-order theories of which the models of the second-order theory are intuitively unwanted), but there are some theories for which it holds. Peano arithmetic is one of them: if a model interprets the natural numbers, it does what we expect it to do. Any model of second-order Peano arithmetic does just that: it interprets a structure isomorphic to that of the natural numbers. We can therefore say that any model of second-order Peano arithmetic is a *standard model.*[102] Since non-standard models are not isomorphic to this intended model, we can label them as *unintended*: they allow for interpretations of Peano arithmetic that do not correspond with our intuitions or with what we meant to codify.[103]

The existence of such non-intended models of first-order Peano arithmetic would imply that first-order languages are not strong enough to interpret only those things we want them to do. More concretely, the existence of non-standard models implies the allowance of first-order theories for models that may codify more than just the natural numbers. These non-standard models have been proved to exist by Thoralf Skolem,[104] and we will sketch the proof of it (without using too many technical details). For more details of the proof, one can refer to Skolem's paper.

Skolem's proof is not exactly like the nowadays commonly used proof using compactness. He does, however, in an elegant way show the existence of non-standard models of Peano arithmetic. In short, he proves that for a set of sentences M such that M contains all first-order formulae definable in $\mathcal{L}$ that are valid for the series of natural numbers N (N thus in fact being a model), we can also define a series $N*$ such that all formulae in M are valid for $N*$, but $N*$ is an extension of N such that it is not the case that $N* \simeq N$.[105]

Concretely, Skolem's proof shows that, using Peano axioms, it is possible to construct a model that contains *more* than just the natural numbers: besides the natural numbers, it contains numbers that are larger than any natural number. It can therefore never be isomorphic to the set of natural numbers N. This clearly proves the existence of non-intended models of Peano arithmetic.

A more modern proof makes use of the compactness theorem.

Definition 8 (Compactness) *If a language $\mathcal{L}$ is* compact, *then any set of formulae Γ is consistent iff every finite subset $\Delta \subseteq \Gamma$ is consistent.*[106]

101. Shapiro, *Foundations without Foundationalism*, 111.
102. Montague, 'Set Theory and Higher-Order Logic', 136.
103. Enayat and Kossak, *Nonstandard Models of Arithmetic and Set Theory*, 1.
104. Skolem, 'Über die Nicht-charakterisierbarkeit der Zahlenreihe mittels endlich oder abzählbar unendlich vieler Aussagen mit ausschliesslich Zahlenvariablen'.
105. Skolem, 159.
106. Halbeisen and Krapf, *Gödel's Theorems and Zermelo's Axioms*, 30.

It is a mathematical fact that first-order logic is compact.[107] We construct a theory based on Peano arithmetic. We abbreviate the first-order Peano axioms as given before by PA. We then define an extension PA^+ with a constant c such that $PA^+ = PA \wedge \forall x(c > x)$. We thus define the interpretation of the constant c to be a numeral larger than any numeral in the standard model of Peano arithmetic. Since Definition 8 holds for first-order logic, if we can show that every finite subset of PA^+ is consistent, we know that also PA^+ as a whole is consistent. By soundness, there then exists a model for PA^+ so that we have shown that there exists a non-standard model of arithmetic, with some number larger than any number in the standard model.

For any finite subset $\mathcal{S}$ of PA^+, we can give a model $\mathcal{M}$ that interprets the non-logical vocabulary in just the same way a standard model of arithmetic would do, except for $c^{\mathcal{M}} = s(n)$ with n the largest numeral of $\mathcal{S}$. So, we define c to be the successor of the largest number of all finite subsets. As $\mathcal{M}$ interprets $\mathcal{S}$ in just the same way as the standard model of Peano arithmetic, we get that $\mathcal{M} \vDash PA$ and since $\mathcal{M} \vDash \forall x \in S(c > x)$ we also get that $\mathcal{M} \vDash \mathcal{S}$. As this construction can be done for all finite subsets of PA^+, we can conclude by compactness that there exists a non-standard model $\mathcal{M}$ of Peano arithmetic.

The fact that such unintended models exist is one of the main reasons for Shapiro to argue against the use of first-order logic to provide a foundation for (parts of) mathematics.[108] Second-order logic, however, is able to characterize the abstract structure of the natural numbers. By characterizing only models isomorphic to the intended model of arithmetic, the second-order Peano axioms single out the *abstract structure* of the natural numbers. Any, and only a, valid interpretation of the natural numbers will satisfy the axioms. More on this will be said in the next part.

There are of course also positive arguments to choose second-order logic to describe mathematics in a logical system. One is epistemic, another one is related to the languages both logical systems make use of. We will have a quick look at them.

The epistemic argument appeals to the idea of why one would accept the induction scheme as used in axiomatizing arithmetic in first-order logic. Shapiro argues that one would accept the induction *scheme* only because one already believes that the induction *axiom* is correct.[109] But in order to do so, second-order logic is already presupposed. If one would reject the second-order axiom because of ontological reasons, using that same axiom to justify the acceptance of the induction scheme is impossible. He states that *if* a non-second-order epistemic justification of the first-order induction scheme exists, it is yet to be found.[110]

It seems that the advocates of first-order and second-order logic use different kinds of arguments while arguing for 'their' kind of logic. First-order logic advocates tend to argue against second-order logic because of its ontological presuppositions, whereas advocates of second-order logic like Shapiro and Kreisel (quoted by Shapiro) give arguments related to epistemics. What counts more is a question that has yet to be answered. From a scientific perspective, epistemological clarity may be preferable. However, preventing a theory from becoming too ontologically 'heavy' might be more important from a philosophical point of view. Shapiro responds to the claim that second-order logic presupposes a too large ontology by stating that the line between logic and mathematics cannot be drawn so clearly;[111] the absence of such a sharp border would imply that logic might presuppose the ontology that is used by mathematics, which is indeed 'staggering', without too many problems.

Another argument that Shapiro gives in favour of second-order logic is related to the expressive power of first and second-order *languages*. It is also closely related to the idea

107. Halbeisen and Krapf, *Gödel's Theorems and Zermelo's Axioms*, 30.
108. Shapiro, *Foundations without Foundationalism*, 116.
109. Shapiro, 118.
110. Shapiro, 118.
111. Shapiro, 97.

of the power of first-order logic to distinguish between different models of arithmetic that we discussed in the paragraph about non-standard models. The point of his argument is that in second-order Peano arithmetic, we do not have to extend our theory with axioms (i) to (iii*) with arithmetical functions like $\times, +$, as those are already derivable from the induction axiom. We can of course explicitly state these functions in our theory, but any model of such a theory will be isomorphic to the models for the more 'bare' theory without extra functions.[112]

This is not the case for first-order logic. Extending, for example, a first-order theory of arithmetic with only the $+$ function to a theory that also has the $\times$ function, can be done in multiple ways. We can, indeed, only extend such a theory non-categorically.[113] This means that an extension of 'basic' arithmetic (so only the definition of the natural numbers) is essentially different from arithmetic with arithmetical functions. Peano arithmetic is relatively simple, but when we want to extend a logical description of the real numbers, we must add terminology for e.g. π, whereas in a second-order theory this kind of special numbers can just be derived.[114] Second-order theories are thus more powerful than their first-order counterparts in the sense that only the basic axioms have to be defined; all other functions and elements can be derived from those. As Shapiro states, in first-order theories the schemes should be modified *on the go*, and it is not a trivial thing to decide when we have added enough of the new functions or elements.

We have now seen some arguments for the use of second-order logic to codify mathematics. Especially the fact that second-order logic is categorical is an important reason to choose for second-order logic rather than first-order logic. As we endorse the structuralist position that structures are to be studied in mathematics, categoricity is of great use. It means that we can actually describe a structure using our logic, without the problem of including unwanted structures in the interpretations of the theory. Hence, we will now continue working in second-order logic.

In the next part, we will spell out the *semantics* for our logic. We will discuss the different types of semantics, and we will give two semantics for second-order logic. Then, we will formalize the notion of *implicit definitions*, which is an account of how formulae acquire their meaning in a structural context. After we have spelled out *how* formulae get their meaning, we will also give a formal account of what that meaning is.

112. Shapiro, *Foundations without Foundationalism*, 120.
113. Shapiro, 121.
114. Shapiro, 122.

Part II

Semantics and Meaning

Up to now, we have mainly focused our research on *intuitive* meaning: When we see and read a formula, we think it has some kind of meaning even though the formula is abstract. The identity of indiscernibles, Formula 3, is a rather clear example of this. We tend to interpret formulae as meaningful sentences, even though strictly speaking, *we* do not give them meaning, but the semantics does.

In a logical system formulae are *given* a meaning by the semantics that is being used in the logic. As we are especially interested in second-order logic, we will look at semantics that are applicable to second-order logic. First, we will distinguish two main types of semantics: proof-theoretic and truth-conditional semantics, sometimes referred to as 'model-theoretic semantics'. We will see that truth-conditional semantics are most useful in our case.

On the level of truth-conditional semantics, we can again distinguish two kinds of semantics. We will look into both full second-order semantics and Henkin semantics, which is very similar to first-order semantics.

After having discussed the two most common semantics in second-order logic, we will continue working in full second-order semantics and investigate the notion of structural definitions. Then, we will develop a view of what the meaning of a formula is in a structuralist context. We argue that the meaning of a formula in such a context is the isomorphism class of structures that satisfy its propositional function.

4 Semantics: Types and Uses

As stated above, there are roughly two different 'kinds' of semantics: proof-theoretic and truth-conditional semantics. We will discuss both and argue that for our goal of understanding the meaning of mathematical second-order formulae, we best choose truth-conditional semantics.

4.1 Types of Semantics, Realism and Intuitionism

The distinction between proof-theoretic semantics and truth-conditional semantics has many philosophical groundings and implications. We will discuss the realism versus anti-realism debate, and the different angles from which the two theories shed light on the question of semantics: from an epistemological or a metaphysical point of view. That question comes down to whether truth is an epistemologically or metaphysically relevant notion.

The notion of proof-theoretic semantics can be linked to the logical basis of the metaphysics of Michael Dummett.[115] According to Dummett, the ultimate question of analytical philosophy is 'what is a theory of meaning?'[116] Both proof-theoretic and truth-conditional semantics give a formal account of a possible answer to this question. When we give a (correct) proof of a formula φ following from a set of formulae Γ, we may be interested in multiple things related to the proof. First, we may be interested in whether the proof holds, so if φ really does follow from Γ. We are then interested in the consequence relation and

115. Schroeder-Heister, 'Proof-Theoretic Semantics'.
116. Dummett, 'What Is A Theory of Meaning? (I)', 1.

we can call this an *extensional* type of semantics. This corresponds to truth-conditional semantics: What the proof actually states does not matter to us, but the consequence relation expressed by the proof does.[117]

We may, however, also be interested in what the proof actually says. In most proof systems, there are many ways to prove that $\Gamma \vDash \varphi$. Even though the relation expressed by the proof is identical in all proofs, the means of arriving there differs. Call this an *intensional* type of semantics, that corresponds to proof-theoretic semantics: We are interested in both the consequence expressed by the proof but also in the way in which it is actually proved.[118]

According to Dummett, adopting a certain semantic view for a given class of statements forces one into accepting a realist or an anti-realist view relative to the objects the statements are about. Accepting a proof-theoretic view of semantics also means accepting an anti-realist view of the objects (and accepting anti-realism for that class of objects also means accepting a proof-theoretic view of semantics), whereas going with truth-conditional semantics forces one into adopting a realist view of the world.[119] Note that Dummett writes '*theory of meaning*' here, but this concept can be perfectly formalized to 'semantics' as well.

This realist or anti-realistist stance to the world not only corresponds to the semantics one chooses to work with, but has an even deeper grounding: the acceptance of the principle of bivalence for a class of statements.[120] The principle of bivalence states that every statement that can be made is either true or false. It is quite reasonable to think that this is not the case for some class of statements: think of ethics or counterfactuals. The sentence 'If the student had not attended the lectures, she would not have passed the course.' does not seem to have a clear truth-value: we cannot determine the truth-conditions of such a sentence, that is, we cannot say what should be the case in the world for the sentence to be true. The same goes for ethical statements: maybe some things are good nor bad.

When we are realist towards a given part of the world, that means that we must also accept bivalence over the class of statements about that part of the world: something *is* the case or it *is not*, thus we can say that some statement about such a *real object* is true or false. This also allows us for adopting a truth-conditional semantics: Some statement may be true or false and a proof can show us that. However, we do not need anything beyond the information *that* the consequence relation of the proof holds; we already know *why* the proof holds: because of the state of affairs in the world.

However, when we adopt an anti-realist view and abandon the principle of bivalence, we cannot justify the truth of our statements by pointing at the state of affairs. A truth-conditional semantics will therefore not fit our needs: we cannot just rely on truth values for our semantics. A proof not only shows that a statement is true, but we can say that the statement is true *in virtue of* the proof we can give for it.[121] Our interest in the proof goes further than just the consequence relation: there are many ways to give a proof for a given statement and if that statement is true in virtue of its proof, it can be true because of different reasons. When we recognize a proof for some statement, we may only know why it is true by having a closer look at the contents of the proof. It does not suffice to point towards the world in order to justify a statement when we are anti-realists. The meaning of a statement is not given by its truth-conditions, but by how a proof for it would be constructed, thus by the inferences and rules used to prove an expression.[122]

Dummett's anti-realist position regarding meaning is heavily inspired by the *intuitionist* position of logic and mathematics. Intuitionistic logic rejects the principle of bivalence,

117. Schroeder-Heister, 'Proof-Theoretic Semantics', 1.1.
118. Schroeder-Heister, 1.1.
119. Dummett, 'What Is A Theory of Meaning? (II)', 64.
120. Dummett, 57.
121. Dummett, 70.
122. Dummett, 70.

which means that both the law of excluded middle ($A \vee \neg A$) and the law of double negation elimination ($\neg\neg A \to A$) are rejected.[123] Proof-theoretic semantics are adapted to these intuitionistic starting points. They do not make use of the law of excluded middle (for example to do *reductio ad absurdum*), and expressions of the form $A \to B$ are interpreted as 'A proof of B can be obtained by a proof of A'.[124]

As stated earlier, the difference between proof-theoretic and truth-conditional semantics can also be put in terms of what we are interested in: metaphysics or epistemology. When we use a proof-theoretic semantics, we are interested in *how* a certain proof goes; we want to *know why* B follows from A. However, when studying truth-conditional semantics, we are solely interested in the truth of a statement, so in the consequence relation a proof proves: *if* B follows from A. In the case of truth-conditional semantics, we are thus interested in how things are: we prove things about the state of affairs in the world and are interested in how the world is. This can be seen as a *metaphysical* point of view.[125] In the case of proof-theoretic semantics though, we are interested in how we know that B follows from A; how the proof actually is constructed. This leads to the view that proof-theoretic semantics is much more epistemological than metaphysical.[126] The state of affairs is not directly related to the truth of a statement, so there is not much to say metaphysically about a proof. A proof might make use the state of affairs to show the truth of a statement, but it is not necessary to do so.

4.2 Truth-Conditional Semantics: an Argument from Incompleteness

Now we have discussed the differences between proof-theoretic and truth-conditional semantics, and have also seen what their groundings and consequences are, we can investigate which of the two we can use best for our main question: what is the meaning of a second-order formula *in a mathematical context*. The question therefore comes down to: what is the best type of semantics to give meaning to a second-order formula in such a context. We will 'choose' between proof-theoretic and truth-conditional semantics here. I will give an argument for the use of truth-conditional semantics *from the incompleteness* of second-order logic.

A semantics ideally is set up in such a way that only the expressions we would *intuitively* call true in the natural language are also true in the intended model given by the semantics. We will call this idea *informal adequacy*. If we use a logical system to codify mathematics, we can state this more narrowed-down: all and only *mathematical facts* are true in the semantics. So, a notion of 'intuitive completeness' would mean that any expression that is a mathematical fact, should be provable in some deductive system.

Definition 9 (Informal adequacy) *A semantics is informally adequate iff any informal mathematical statement A is true according to the mathematical theory iff the formalization φ of A is true in the intended model given by that semantics.*

Let us again focus on second-order Peano arithmetic. We have seen that we can categorically define Peano arithmetic in second-order logic with standard semantics and that the second-order theory of arithmetic therefore is semantically complete. But we have also seen that any theory that codifies arithmetic is negation *incomplete*: there are statements that can neither be proved, nor can their negation. This yields the theorem that second-order

123. Iemhoff, 'Intuitionism in the Philosophy of Mathematics'.
124. Sundholm, 'Proof Theory and Meaning', 485.
125. Schroeder-Heister, 'Proof-Theoretic Semantics', 1.1.
126. Schroeder-Heister, 1.1.

logic is incomplete. But it also gives us an important hint about which type of semantics we should use.

The categorical formulation of Peano arithmetic in second-order logic with standard semantics is one in which the 'only' model (all models that exist are identical up to isomorphism) actually is the *intended* model: all and only statements that we would intuitively think should be true, are true in the model. The model of the theory is the model of the natural numbers – exactly what it should be. We can therefore say that the model-theoretic approach of second-order Peano arithmetic narrows down all and only the natural numbers. The structure that is given by the model covers everything we want it to cover, *and nothing more*. Moreover, we can tell the truth-value of the formalization of any statement about the natural numbers – second-order Peano arithmetic is semantically complete after all.

The intuitions we have about the natural numbers are thus formalized in the model given by the standard semantics of second-order logic when applied to arithmetic. Standard semantics can account for an *informally adequate* formalization of arithmetic. Any other approach would need to satisfy the principle of informal adequacy as well to qualify as a good semantics for second-order logic in mathematical contexts. We will see that proof-theoretic semantics cannot guarantee that our intuitions are formalized in the right way.

As said, we want a semantics to formalize the intuitions we have about a certain (part of a) language or theory. A proof-theoretic semantics should eventually yield the same truths in a model, consequences and validities as the truth-conditional semantics to be informally adequate. Again applied to Peano arithmetic, this means that any truth that holds in the intended model should be provable, so that it also hold in the proof-theoretic semantics.

We have seen, though, that second-order logic is *incomplete*; there are true statements according to the standard semantics, that we cannot prove in any deductive system. For a statement to be rendered true by a proof-theoretic semantics, it must be possible to *prove* it. Since proving a statement semantically in order to check its truth-value in that semantics is not viable, this must be done deductively.

But if there are some true statements about a theory that we cannot prove, this means that any proof-theoretic semantics is intuitively incomplete. That is, there are statements that *should* be valid in the semantics, but which are not: they cannot be proved. That yields the result that some statements that are true according to the standard semantics (that yield an intended model) are not true in a proof-theoretic semantics. This can best be seen by using the Gödel sentence as an example.

The Gödel sentence G is a sentence which refers to itself by stating its own unprovability. G: 'G is not provable in any ω-consistent formal system F'.[127] Of course the sentence is, in case of Peano arithmetic, given in terms of arithmetic, but it amounts to the same: it asserts that the sentence itself is not provable. If we can prove that this is true, this leads to a contradiction. If we can prove that the negation of G is true, that would mean that the sentence *is* provable which again leads to a contradiction. The Gödel sentence thus is provable nor refutable.[128]

But the fact that the sentence is provable nor refutable means that G is *true*. In truth-conditional semantics, we can account for that. The properties of the Gödel sentence are such that the truth-conditions of the sentence are fulfilled. However, in proof-theoretic semantics we face a problem: if we cannot *prove* a sentence, we cannot tell whether it is true or not.

This means that proof-theoretic semantics can never account for the statements we think must be true. The model theory and the proof-theoretic semantics do not 'rhyme', which is problematic because we intuitively feel that the standard semantics yields the correct,

127. Kennedy, 'Kurt Gödel', 2.2.2.
128. Kennedy, 2.2.2.

intended model. That also means that no proof-theoretic semantics will ever be able to match this intended model: there *will* be unprovable but intuitively true statements, due to Gödel's incompleteness theorems. Hence, proof-theoretic semantics will, in the case of second-order logic, always be intuitively inadequate.

It would therefore be a bad idea to use a proof-theoretic semantics for second-order logic: doing so would mean that not all *intuitively* true statements can be formalized in the logical system. A truth-conditional theory of meaning does not have this drawback: we have already seen that standard semantics yields the intended model for, for example, Peano arithmetic.

4.3 Truth-Conditional Semantics: an Argument from Realism

There are more reasons to prefer truth-conditional over proof-theoretic semantics. The argument from incompleteness gives a very clear result, but also has implications for our position in terms of the realism–anti-realism debate. I will argue why from a (structural) realist perspective it is necessary to use a truth-conditional theory of meaning. This is an argument coming from a realist position. Of course, assuming realism does weaken the argument, but we have already seen that there is another reason to choose truth-conditional semantics. Moreover, in our eventual research question, finding the meaning of a formula describing a structure, the realist assumption is already made. We then thus have two compelling reasons to prefer truth-conditional semantics over proof-theoretic semantics, of which one applicable to any context in which second-order logic is used.

We will argue that it is natural to accept a truth-conditional semantics when being realist with respect to that part of the world that the theory of meaning covers. Therefore, we should first have made clear why a realist position implies a truth-conditional theory of meaning. We will then lay out why realism with respect to structures is a reasonable starting point.

Proof-theoretic semantics is inherently inferential, as proofs are inferential: they make use of inferences to constitute their meaning.[129] So, the meaning of an expression depends on the inferential role it plays in a proof. More concretely, the meaning of an expression depends on the proof one can give for it.

Suppose that we are realists towards the natural numbers: we think that '$1 + 1 = 2$' is true because there exists an object referred to by '1', which equals an object referred to by '2' when it is added to itself. In a truth-conditional theory of meaning, this idea can be very well embedded: if '1' denotes the object 1 and '2' denotes the object 2, then the meaning of '$1 + 1 = 2$' is given by its truth-conditions, which are the state of affairs regarding the numeral objects.

However, if we adopt a proof-theoretic semantics, we could not sustain our realist position in a meaningful way. We would actually 'lose' the reference to the world by adopting such a semantics: There is, in 'bare proof-theoretic semantics', no way in which we can guarantee that the meaning of an expression '$1 + 1 = 2$', that is the proof for the expression, actually is about the objects of which we think that they exist. The inferences used in the proof do not necessarily make use of reference to the actual objects out there. It might be that the proof holds because of certain properties of the object 1, but it is not necessarily the case that the justification of the proof uses reference to that object.

Although there is no *contradiction* in being realist and accepting a proof-theoretic semantics, the tension is clear: the realist position loses its *actual realism* if the meaning of expressions does not refer back to the objects regarding to which one is realist. The expressions about these objects are not necessarily rendered true by the state of these objects;

129. Schroeder-Heister, 'Proof-Theoretic Semantics', 1.1.

one could as well accept an anti-realist position then. A truth-conditional theory of meaning appears to be a more natural choice, given a realist position towards mathematics.

There is more to say about this. It is not very difficult to find a way of grounding one's proof-theoretic theory of meaning such that proofs are ultimately grounded by the objects the relevant statements are about. If that approach works, it would still be possible, and result in less friction, to adopt a proof-theoretic semantics while insisting on a realist metaphysical view. For example, one could maintain that the rules and axioms that are used in a proof-theoretic semantics are trivially true because they are *made true* by the objects they are about.

However, this leaves two problems unsolved. First of all, there is still the problem of incompleteness as discussed before. Moreover, because of proof-theoretic semantics being a formalization of the *meaning is use-principle*,[130] proof-theoretic semantics contains principles of pragmatism that are unacceptable from a realist perspective.

A theory of inferentialism (which also applies to proof-theoretic semantics) contains a lot of pragmatism, due to it being a formalization of the principle of *meaning is use*.[131] This implies that meaning not only depends on the inferences made in order to write a proof for a certain expression, but also on the *ability of a speaker* to make those inferences.[132] Thus, if a certain proof has not yet been found, but in principle can be given, a statement such as $1 + 1 = 2$ may still not be true. This is of course problematic from a realist perspective, as the ability of drawing inferences does not depend or only partly depends on the actual state of affairs. The meaning of a statement becomes an epistemological notion (whether one can draw the relevant inferences or not) instead of a metaphysical one (whether certain objects are so and so).

4.4 Structural Realism and Abstract Structure

So, if a realist position makes sense in our case, this is another reason to accept a truth-conditional theory of meaning as an alternative to a proof-theoretic semantics. If we have another look at the ultimate goal of this thesis, we will see that a realist position towards the second-order expressions is useful and justified. To do so, we will distinguish three structuralist positions: *ante rem*, *in re*, and *post rem* structuralism.[133] We will see that, by the question posed in the thesis, ante rem structuralism is a natural position to choose here and corresponds to a realist position regarding structures.

The positions of *in re* and *post rem* structuralism both state that structures do not exist independently of the systems that individuate them.[134] *In re* structuralists think that structures exist in the mathematical systems that exemplify them,[135] whereas *post rem* structuralists state that the relevant structures are built out of the elements found in a system, thus being posterior to the mathematical systems.[136]

According to *ante rem* structuralism, sometimes called *ante rem realism*, the abstract structures of mathematical systems exist prior to those systems.[137] Although all three positions can be realist, only ante rem structuralism is realist towards *abstract structures*; the other positions admit the existence of structures only in an individuated sense: without at least one system that individuates a structure, the structure would not have existed.

130. Schroeder-Heister, 'Proof-Theoretic Semantics', 1.1.
131. Kügler, 'Putting Brandom on His Feet', 79.
132. Kügler, 80.
133. Reck and Schiemer, 'Structuralism in the Philosophy of Mathematics', 2.3.
134. Shapiro, *Philosophy of Mathematics*, 84.
135. Shapiro, 85.
136. Reck and Schiemer, 'Structuralism in the Philosophy of Mathematics', 2.3.
137. Shapiro, *Philosophy of Mathematics*, 84.

Concretely: the natural numbers structure would not have existed if there was no system individuating the structure of the natural numbers (whatever such a system might be). Dummett puts ante rem structuralism as 'mystical structuralism', the position according to which mathematics is concerned with *abstract structures*, so independent of their exemplifying systems.[138]

Mathematicians are interested in structural properties: it does not matter what the properties of the actual objects are in a system of the natural numbers, as two instantiations of the natural number 2 have the same structural properties. We can therefore state that what is of interest is the *abstract structure* of, in this case, the natural numbers. Since second-order logic can describe such abstract structures so that we can study them, it would be a natural thing to accept an ante rem structuralist view. Not only is this view very perspicuous, as it is clear on how structures in general work and on how structures exist in mathematical systems, it is also a reasonable assumption to say that the structures of which we wonder *what they are*, exist: It makes much more sense to ask for the ontological status of something we think is real in this case. Ante rem structuralism, according to Shapiro, also comes closest to capturing *how mathematical theories are conceived.*[139] The idea of studying abstract structures as they are, so without being invoked by a system, is called *free-standingness* by Shapiro.[140] We can make statements about the natural numbers that are true of any system that exemplifies the natural number structure. The structure then really is *freestanding*: it does not matter in what way the places in the structured are occupied, as long as they bear the right relations to each other. We can talk of 'the number three', but what we really mean is the object that plays the role of three in any natural number system. We can talk about structures without knowing, or even caring about, the concrete systems that exemplify these structures. The ante rem account of structuralism can accommodate these insights about structure in a very clear way,[141] which is why we can reasonably make it an underlying premise in this thesis.

By having good reasons to accept ante rem structuralism, we also have good reasons to accept a truth-conditional theory of meaning. We will therefore continue by defining two different truth-conditional semantics for second-order logic.

The position of structural realism does raise other meaning-related questions: How are logical expressions and structures formally related? How is a formula given meaning by the structural context? We have seen that logical expressions (axiomatizations) can characterize a structure, without actually invoking a model-theoretic instance of that particular structure. In a way, the expression seems to *define* the structure, without explicitly doing so. There exists a theory endorsed by structuralist that accounts for this: *implicit definition.* We will discuss this theory, after we have given a brief overview of second-order truth-conditional semantics.

4.5 Second-order and Henkin Semantics

We will distinguish two different types of semantics for second-order logic: standard second-order semantics and Henkin semantics. The latter 'weakens' second-order logic to regain some of the features of first-order logic that are lost by second-order logic, such as completeness.[142] However, Henkin semantics also has its drawbacks: one cannot categorically define infinite structures in it.[143]

138. Dummett, *Frege*, 295.
139. Shapiro, *Philosophy of Mathematics*, 90.
140. Shapiro, 100.
141. Shapiro, 100.
142. Shapiro, *Foundations without Foundationalism*, 90.
143. Shapiro, 95.

We will briefly discuss the technicalities of both types of semantics, after which we will see that only standard second-order semantics can suit our needs, especially categoricity. We will discuss a few properties of both semantics: the definition of a model, validity, satisfiability and consequence. The definitions below come from chapter 3, Shapiro (2001).[144] More details can be found there.

First, we will look at full second-order semantics. Like in first-order logic, the most important notions of our semantics are the *model* and the *assignment function*. The model $\mathcal{M}$ consists of a pair $\langle \mathcal{D}, \cdot^{\mathcal{M}} \rangle$, in which $\mathcal{D}$ is the domain of discourse and $\cdot^{\mathcal{M}}$ is the interpretation function that assigns members of $\mathcal{D}$ to any non-logical item of our language (think of constants and predicates). The assignment function α assigns to each first-order variable a member of $\mathcal{D}$, and to each n-ary predicate variable a subset of $\mathcal{D}^n$. Each n-ary function variable gets assigned a function from $\mathcal{D}^n$ to $\mathcal{D}$. The set of all variables, functions and constants is called the set of *terms*.

We then want to know how we should interpret second-order variables, both free and bound. We distinguish the following cases:

Free variable: If X^n is a relation variable and $\langle t \rangle_n$ is a sequence of n terms, then $\mathcal{M}, \alpha \vDash X^n \langle t \rangle_n$ iff the sequence of members of $\mathcal{D}$ denoted by $\langle t \rangle_n$ is an element of $\alpha(X^n)$.

Universal quantifier: $\mathcal{M}, \alpha \vDash \forall X \varphi$ iff $\mathcal{M}, \alpha' \vDash \varphi$ for every assignment α' that agrees with α at every variable except possibly X.

Existential quantifier: $\mathcal{M}, \alpha \vDash \exists X \varphi$ iff $\mathcal{M}, \alpha' \vDash \varphi$ for some assignment α' that agrees with α at every variable except possibly X.

Function variable: $\mathcal{M}, \alpha \vDash \forall f \varphi$ iff $\mathcal{M}, \alpha' \vDash \varphi$ for every assignment α' that agrees with α at every variable except possibly f.

We now only need to define the notions of validity, satisfiability, and consequence to get a good grasp of standard semantics. We say that φ is valid iff $\mathcal{M}, \alpha \vDash \varphi$ for every $\mathcal{M}, \alpha$. Γ is satisfiable iff for some $\mathcal{M}, \alpha$, $\mathcal{M}, \alpha \vDash \varphi$ for every $\varphi \in \Gamma$. φ is a consequence of Γ iff $\Gamma \cup \{\neg\varphi\}$ is not satisfiable. These notions are the same as in first-order logic.

Henkin semantics narrows down the expressive power of second-order logic by restricting the range of the second-order variables. It does that by having a *fixed collection* of relations on the domain, over which the relation variables can range.[145] The definitions below again come from chapter 3, Shapiro (2001). More details can be found there.

A Henkin model $\mathcal{M}^H$ consists of a tuple $\langle \mathcal{D}, R, F, \cdot^{\mathcal{M}^H} \rangle$ with $\mathcal{D}$ the domain of discourse, $\cdot^{\mathcal{M}^H}$ the interpretation function like in standard semantics. There are two new elements in $\mathcal{M}$: R is a sequence of sets of relations and F is a sequence of sets of functions. For each number n, $R(n)$ is a (non-empty!) subset of the powerset of $\mathcal{D}^n$. That is, for any arity n, $R(n)$ gives a fixed collection of relations over $\mathcal{D}$. $F(n)$ on its turn gives a non-empty set of functions from $\mathcal{D}^n$ to $\mathcal{D}$. Both R and F thus give a *fixed range* to both relation and function variables.

It will be no surprise, then, that variable-assignment function α assigns a member of $R(n)$ to every n-ary relation variable, and a member of $F(n)$ to every n-ary function variable.

The interpretation of the second-order variables remains the same. The real difference lies in the assignment function: instead of ranging over the complete domain, second-order variables are bound to a subset of the domain. Of course it is possible to think of an $\mathcal{M}^H$

144. Shapiro, *Foundations without Foundationalism.*
145. Shapiro, 73.

with R and F such that $R(n)$ equals the powerset of $\mathcal{D}^n$ and $F(n)$ is the set of *all* functions from $\mathcal{D}^n$ to $\mathcal{D}$. We then say that $\mathcal{M}^H$ is a *full model*.

As has been stated earlier, any second-order language with Henkin semantics cannot categorically characterize a structure. Since we are interested in the meaning of expressions that categorically define such structure, Henkin semantics is not adequate for our goals. We will therefore stick with standard second-order semantics. As deductive systems are not our main interest (meaning is), we can live without completeness without any problems.

Also note that Henkin semantics share many features with first-order logic semantics. In fact, there exist implementations of first-order semantics for second-order logic that, like Henkin semantics, bring back completeness, but at the cost of categoricity.[146] First-order semantics and Henkin semantics are actually equivalent: For every first-order model there exists a Henkin model such that a sentence φ is true in that model iff it is true in the first-order model. Henkin semantics and first-order semantics thus are essentially the same, despite a few technical differences.[147]

Before moving to the meaning of formulae in mathematical contexts, we need to discuss how their meaning can be determined. We will do so by laying out the ideas of *implicit definition* in the next section.

5 Implicit Definitions and Structuralism

We have seen that it is possible to give an account of several mathematical structures by giving second-order axioms. One of those structures is the natural-number structure: Using the Peano axioms, we can categorically characterize arithmetic and therefore the structure of the natural numbers.

However, the link between those structures and their (second-order) axiomatization is still unclear. How do these formulae in mathematical contexts acquire their meaning? Structuralists have a common answer to this: implicit definitions. Since there already exists a theory of how the meaning of a statement in a mathematical context is determined, it can help us with determining *what* the meaning is, as there is not yet a satisfying answer to this question.

Giovannini and Schiemer give an informal definition of implicit definition:

Definition 10 (Implicit definition) *A theory* implicitly defines *predicate R iff it* uniquely *determines its interpretation relative to a given interpretation of the non-logical symbols of the base language.*[148]

Applied to a structure, such as the natural numbers, we can say that the facts that (i) zero is not the successor of any number, (ii) every natural number has a unique successor and (iii) the induction principle holds, the Peano axioms, together *implicitly define* a predicate $\varphi_N(x)$: 'is a natural number', and therefore uniquely determine the structure of the natural numbers.

We will work out Giovannini's and Schiemer's more formal account of implicit definition[149] a bit more. We can give a formal account of this definition as follows. The formula $\varphi(\vec{x})$ is a formula φ possibly with free variables $x_1, \ldots, x_n$. $\vec{x}$ thus stands for the variables

146. Shapiro, *Foundations without Foundationalism*, 95.
147. Shapiro, 76.
148. Giovannini and Schiemer, 'What are Implicit Definitions?', 4.
149. Giovannini and Schiemer, 3.

$x_1, \ldots, x_n$. Note that $\varphi(\vec{x})$ is a complex predicate, consisting of formulae, such that the interpretation of $\varphi(\vec{x})$ in a model $\mathcal{M}$ is the set consisting of exactly these objects for which $\varphi(\vec{x})$ holds according to $\mathcal{M}$. So, with $\mathcal{M}, \alpha \vDash \varphi(\vec{d})$ we mean $\mathcal{M}, \alpha(\vec{x} \mapsto \vec{d}) \vDash \varphi(\vec{x})$. Note that $\vec{d} = (d_1, \ldots, d_n)$ here.

$$\varphi(\vec{x})^{\mathcal{M}} = \{\vec{d} : \mathcal{M} \vDash \varphi(\vec{d})\}$$

Another terminological note has to be made on the interpretation of a theory $\mathbb{T}$ in a model $\mathcal{M}$. The interpretation of the theory is an n-tuple consisting of the interpretation of the constants, the function symbols and the predicates that are being used in the theory. Basically, this is a model without the domain: $\mathbb{T}^{\mathcal{M}}$ is the interpretation function applied to all non-logical vocabulary in $\mathbb{T}$, but the domain still has to be given.

$$\mathbb{T}^{\mathcal{M}} = (c^{\mathcal{M}}, f^{\mathcal{M}}, R^{\mathcal{M}} : c \in \mathcal{C}, f \in \mathcal{F}, R \in \mathcal{R})$$

Here $\mathcal{C}$ is the set of constants in $\mathbb{T}$, $\mathcal{F}$ the set of functions and $\mathcal{R}$ the set of predicate symbols. The formal definition for implicit definitions then is:

Definition 11 (Implicit definition) *A complex predicate $\varphi(\vec{x})$ is implicitly defined by a theory $\mathbb{T}$ iff for all models $\mathcal{M}, \mathcal{N}$, if $\mathbb{T}^{\mathcal{M}} = \mathbb{T}^{\mathcal{N}}$, then $\varphi(\vec{x})^{\mathcal{M}} = \varphi(\vec{x})^{\mathcal{N}}$.*

This matches with the informal Definition 10: Relative to the interpretation of the non-logical vocabulary of $\mathbb{T}$, the interpretation of predicate $\varphi(\vec{x})$ is fixed. If for any two models, the interpretation of $\varphi(\vec{x})$ is equal if and only if the interpretation of $\mathbb{T}$ is equal, we may state that it is fixed relative to $\mathbb{T}$.

We have seen in Part 1 that the meaning of a predicate in a model is determined by giving its extension. In this case, we could use as a domain the actual numbers, or any other set that has the same structure. We must, however, fix it in advance. If we then give the extension d as the interpretation of $0^{\mathcal{M}}$, and interpret successor function $S^{\mathcal{M}} = f : \mathcal{D} \to \mathcal{D}$, then the extension of $\varphi_N(x)$ follows from the two interpretations we have just fixed and it becomes $\varphi_N(x)^{\mathcal{M}} = \{d, f(d), f(f(d)), \ldots\}$. So, once the interpretations $0^{\mathcal{M}}$ and $S^{\mathcal{M}}$ are fixed, the extension of $\varphi_N(x)$ follows.

The predicate $\varphi_N(x)$ is thus *implicitly* defined by implicitly giving its extension, which can be inferred from the axioms (i), (ii), (iii): The axioms define which objects in the model satisfy the formula. We will see that this approach is not feasible from a structuralist perspective. In the next section, we will discuss the extension of implicit definitions that define *abstract* structures. We will also say more on this in the next chapter, in which we discuss propositional functions.

5.1 Structural Definitions and Abstract Structures

By fixing the meaning of the successor function S and the number zero, the predicate $\varphi_N(x)$ is defined, and its meaning is uniquely determined. Hence, this is an implicit definition according to Definition 11. However, by fixing the meaning of 0 and successor function S, the implicit definition is still limited to a particular system. The facts that are stated by the Peano axioms are purely *structural*, they only say things about the relations that the natural numbers bear to each other. The objects themselves are not relevant here, only structure matters.[150] It would therefore be better if we could make use of a type of definition that does not need a fixed interpretation of the non-logical vocabulary.

Structural accounts of implicit definitions, defining abstract structures, are dubbed *structural definitions* by Giovannini and Schiemer.[151] This distinction requires a slightly different

150. Shapiro, *Philosophy of Mathematics*, 130.
151. Giovannini and Schiemer, 'What are Implicit Definitions?', 4.

definition: the interpretation of non-logical symbols does not have to be already fixed. That yields the following informal account of structural definitions:

Definition 12 (Structural definition) *A theory* structurally defines *predicate R iff it* uniquely *determines its interpretation.*[152]

Following Giovannini and Schiemer, we rename the implicit definitions as in Definition 10 as *implicit definitions in the strict sense*. It can be the case that some terms that are used in the structural definition by a theory, are defined *themselves* in that theory.[153] It is necessary then, however, that the theory is *coherent*,[154][155] such that the theory singles out a structure by giving the axioms the structure should satisfy.

Because Giovannini and Schiemer do not make the difference between these two types of implicit definitions extremely clear, we will point out here why distinguishing them is important. We are particularly interested in definitions not of concrete systems, but of abstract structures. By fixing the meaning of all non-logical vocabulary except the defined predicate, as one does in an implicit definition in the strict sense, one already fixes one system. In the example of arithmetic, the constant 0 is already fixed to refer to a certain object; the definition determines a system, not an abstract structure.

This is exactly why implicit definitions in the strict sense will not do for structuralists. Arithmetic, for example, has many different instantiations which are not identical, but which are isomorphic. Two interpretations should both fit the definition of arithmetic if they bear the same structure. However, the interpretation of 0 and successor function S, as well as the domain $\mathcal{D}$, may be different. To successfully define the *abstract* structure of the natural numbers, one should therefore not fix the non-logical vocabulary to any particular interpretation. The underlying idea of determining the meaning of a predicate as determining its extension then does not work any more: by determining the extension, one fixes the interpretation. The meaning of a predicate should therefore be given in other terms, which accounts for the idea of a structure being defined, and the possible existence of many instantiations that bear that structure (thus are isomorphic).

If the meaning of the non-logical vocabulary is not yet fixed, but all non-logical symbols can be specified by the same implicit definition, as can be done in structural definitions, the definition is not tied to any system in particular. In case of arithmetic, we would get just the axioms (i), (ii) and (iii). When a system bears the right structure, thus is isomorphic to the natural number structure, it fits the definition. The *structural* definition of Peano arithmetic pins down any system that is isomorphic to the intended model. In fact, it yields essentially only one structure, as second-order Peano arithmetic is categorical. However, the definition is not limited to one system.

The important difference between fixing the meaning of non-logical vocabulary or not is the ability to single out an abstract structure or only concrete systems respectively. It is also important to note that, when defined structurally, $\varphi_N(x)$ is not a 'primitive predicate' any more in the sense that its extension can simply be given in terms of domain members. There is not an extension one could give as there is no model to which $\varphi_N(x)$ is relatively defined. $\varphi_N(x)$ stands for a formula that singles out the right structures: The complex predicate should be true for all systems that exemplify the natural number structure.

With this knowledge at hand, we can give a formal account of structural definitions. First, we again need to get some terminology straight. First of all the notion of *substructure*:

152. Giovannini and Schiemer, 'What are Implicit Definitions?', 4.
153. Giovannini and Schiemer, 4.
154. Shapiro, *Philosophy of Mathematics*, 133.
155. Please note that Giovannini and Schiemer use the term 'consistent' here. I choose to use the, admittedly more vague, term 'coherent' here due to incompleteness issues. We will come to coherence later.

any model $\mathcal{M}$ that interprets a theory $\mathbb{T}$, is a structure for that theory. We can define *substructures* of such a model for any given (complex) predicate or formula.

Definition 13 (Substructure of a model) *A structure $\mathcal{S}$ is a* substructure of $\mathcal{M}$ *iff $\mathcal{S}$ interprets just a subset of the non-logical vocabulary that $\mathcal{M}$ interprets such that for all σ interpreted by $\mathcal{S}$ it is the case that $\sigma^{\mathcal{S}} = \sigma^{\mathcal{M}}$.*

Definition 14 (Model for a formula) *A structure X, written $\mathcal{M}_\varphi$, is a* model for a formula $\varphi(\vec{x})$ *iff $\mathcal{M}_\varphi = (\varphi(\vec{x})^{\mathcal{M}}, c^{\mathcal{M}}, f^{\mathcal{M}}, R^{\mathcal{M}} : c \in \mathcal{C}_\varphi, f \in \mathcal{F}_\varphi, R \in \mathcal{R}_\varphi)$.*

Note that with $\mathcal{C}_\varphi$ we mean $\mathcal{C}_\varphi = \{c \in \mathcal{C} : \text{ c occurs in } \varphi(\vec{x})\}$. The definitions of $\mathcal{F}_\varphi$ and $\mathcal{R}_\varphi$ are analogous.

So, the model $\mathcal{M}_\varphi$ for a formula $\varphi(\vec{x})$ consists of an interpretation of just the non-logical vocabulary that occurs in $\varphi(\vec{x})$. It interprets the non-logical vocabulary in the same way as some (possibly larger) model $\mathcal{M}$. That means that $\mathcal{M}_\varphi$ is a submodel of $\mathcal{M}$.

Theorem 1 (Submodel for a formula) *If $\mathcal{M}_\varphi$ is a model for some formula $\varphi(\vec{x})$, then it is a* submodel *of a possibly larger model $\mathcal{M}$.*

Note that the *domain* of some $\mathcal{M}_\varphi$ is the extension of $\varphi(\vec{x})$ in that model, that is, the set of things that make $\varphi(\vec{x})$ true. In case of two models the domains may be different, but will still be isomorphic. Concretely, the domain of $\mathcal{M}_\varphi$ consists of the elements necessary to interpret $\varphi(\vec{x})$ so that $\varphi(\vec{x})$ is made true: $\mathcal{D} = \{\vec{d} : \mathcal{M}_\varphi \vDash \varphi(\vec{d})\}$. Now we have the terminology set for the formal definition of *structural definitions*:

Definition 15 (Structural definition) *A complex predicate $\varphi(\vec{x})$ is structurally defined by a theory $\mathbb{T}$ iff for all models $\mathcal{M}, \mathcal{N}$, if $\mathcal{M} \vDash \mathbb{T}$ and $\mathcal{N} \vDash \mathbb{T}$, then there exists a substructure $\mathcal{M}_\varphi$ and $\mathcal{N}_\varphi$ such that $\mathcal{M}_\varphi \simeq \mathcal{N}_\varphi$.*

As we do not have to fix the non-logical vocabulary, as there are many models that satisfy $\mathbb{T}$ and so there are many models that interpret $\varphi(\vec{x})$ in a different way, the extension of $\varphi(\vec{x})$ is not limited to a specific system any more. In the case of arithmetic, any model that satisfies the Peano axioms will eventually interpret a structurally defined predicate in such a way that the interpretations between different systems are isomorphic. An isomorphism between interpretation here means: a bijection from the one substructure-interpretation to the other.

This relates to what Shapiro calls the *places-are-objects*-view of structuralism, in contrast with treating the places in a structure as *offices.* According to the latter position, we should think of places in a structure in terms of systems that exemplify the structure, whereas according to the position of places-are-objects, the places in an abstract structure should be treated as actual objects.

We find a good real-life example in the work of Shapiro, making use of the political system in America. If we think of the different places in the structure of America's politics as offices, we see them as places that yet have to be filled by an object. The office of the vice-president may be filled by Mike Pence or by Kamala Harris, and we can talk about differences between two vice-presidents: The one may be more intelligent than her or his predecessor. When we talk about the *vice-president* we mean an office that is filled by a concrete person: We discuss the places in the structure in terms of an exemplification of it. So, we can talk about different natural number systems that are isomorphic, but implement the structure in different ways. That makes the statement that 'the von Neumann 2 has one more element than the Zermelo 2' is a meaningful one.[156] The idea of strict implicit definitions and of determining the meaning of a predicate as determining its extension fit this places-as-offices perspective. For *ante rem* structuralism, however, we need a different perspective: We need to be able to discuss structures without having to fix extensions of non-logical vocabulary, such that many isomorphic structures fit the definition we give.

According to the places-are-objects-position, we should treat positions in a structure as actual objects. When we say that the vice-president of America is president of the Senate, we do not make a statement about one concrete vice-president, but about the structure as a whole. *Any* vice-president is president of the Senate, since that is how the relations in the structure are. The same can be said of arithmetic: Arithmetic is about the structure of the natural numbers, and 'number 2' refers not to an actual system, but rather to the second place in the structure. The statement about the number of elements in 2 does not make much sense, according to this view. The statement that '2 is the only even prime number' does: We refer not to a specific system, but to a place in the natural number structure in general.[157]

What can we then call the meaning of the 'meaning' of a predicate? The meaning of a structurally defined predicate cannot be determined by its extension, but it has to account for the many different isomorphic systems. What we ultimately want, is that the meaning of the example predicate $\varphi_N(x)$ 'is a natural number' contains *all* systems that satisfy the structure. Both $\varphi_N(\{\{\},\{\{\}\}\})$) (the Von Neumann ordinal 2) and $\varphi_N(\{\{\{\}\}\})$ (the Zermelo ordinal 2) should be true, even though the interpretation of the other non-logical vocabulary is different between the two systems. Giovannini and Schiemer state that a natural model-theoretic framing of a structural definition would be a class of structures that satisfy the axioms.[158] We will work out a model-theoretic account in terms of *isomorphism classes*.

An *isomorphism class* of a structure $\mathcal{M}$ consists of all structures that are isomorphic to $\mathcal{M}$. Hence, they share the same abstract structure. In the case of a predicate, the extension of a structurally defined predicate would be the class of all interpretations by a model that is isomorphic to the abstract structure defined by the structural definition. The *meaning* of a predicate is not simply a fixed extension, but a *class of interpretations, determining (possible) extensions*. If predicate $\varphi_N(x)$ is defined by a structural definition, then the meaning of $\varphi_N(x)$ in model-theoretic terms would be the class consisting of all interpretations by models of the natural numbers. These models may have a different domain and interpret $\varphi_N(x)$ in a different way, but their structure eventually is identical, so that the interpretations of $\varphi_N(x)$ are isomorphic as well. The extension of a structurally defined predicate $\varphi_N(x)$ thus is the isomorphism class of some domain and interpretation $(\mathcal{D}, \varphi_N(x)^{\mathcal{M}})$ of a model $\mathcal{M}$ that satisfies the definition.

As we already said, the domain of a substructure $\mathcal{M}_\varphi$ for some formula $\varphi(\vec{x})$ is the extension of $\varphi(\vec{x})$ *in that model*. A substructure of a model of arithmetic for some formula defined by arithmetic thus has a domain that is equal to the extension of $\varphi(\vec{x})$ in that model, such that the domain and interpretation occur in the isomorphism class of $\mathcal{M}_\varphi$. All models that satisfy the theory by which $\varphi(\vec{x})$ is structurally defined interpret $\varphi(\vec{x})$ in structurally the same way. That aligns nicely with the idea of the extension of a structurally defined predicate being an isomorphism class. Since all substructures $\mathcal{M}_\varphi$ that satisfy $\varphi(\vec{x})$ must be isomorphic if $\varphi(\vec{x})$ is defined by a categorical theory, the extension of $\varphi(\vec{x})$ should contain all structures containing the interpretation and the domain with all objects to which the interpretations of the non-logical vocabulary could possibly refer. With an isomorphism class of models this is accomplished. We argue in the next section that a good way to determine whether a structure satisfies a formula, is to determine whether that structure makes the

156. Shapiro, *Philosophy of Mathematics*, 82.
157. Shapiro, 83.
158. Giovannini and Schiemer, 'What are Implicit Definitions?', 21.

propositional function of the formula true. First, we will cover coherence and categoricity as requirements for a successful structural definition.

5.2 Coherence and Categoricity

In order to successfully single out such a structure, two requirements are in place: first of all, the structure must exist (otherwise nothing is to be singled out by the axioms). Secondly, the definition should only single out a *unique* structure; if a definition fails to uniquely define an object, it is ambiguous which is problematic if one tries to be as precise as mathematics demands one to be.[159] In model-theoretic terms, we could say that a structural definition defines a *class of models*: The axioms of Peano arithmetic structurally define the class of all natural number systems (thus being isomorphic to the natural number structure).[160] By understanding structural definitions in that way, we can talk about abstract structures without having to refer to concrete systems that exemplify them.[161] After all, the axioms extensionally define a *class of models*, which could be seen as possible instantiations,[162] but do not make use of any of them. In the case of a categorical theory, only one structure is characterized.[163]

We now need to make clear how we can make sure a structural definition actually defines an existing structure and its uniqueness. We will make use of two crucial concepts: coherence and categoricity. Put differently, we want a structural definition to define *at least* one structure and *at most* one structure.[164]

As we have already discussed categoricity, the uniqueness property is easiest to discuss first. We have already seen that there sometimes exists a relation between second-order theories and their models, such that any model eventually comes down to being the same as any other model. First-order theories do not have this property: We have seen that there exist non-standard models of arithmetic. The same goes for other first-order theories with infinite models. Unintended models cannot be ruled out, as the Skolem paradox shows us.[165]

First-order theories can thus not be said to *uniquely* determine a certain structure. There exist non-standard models of for example arithmetic, that do not bear the same structure as the intended model: the two structures are not isomorphic. However, second-order logic *can* uniquely determine structures. By giving a *categorical* second-order theory, we can make sure that the determined structure is unique.

All models of a categorical theory are isomorphic, and therefore share the same structure. Hence, only one structure is determined. There may be many models, but all those models essentially are identical. If a categorical theory characterizes a structure, it uniquely does so. Structural definitions in second-order languages (of infinite structures that can be categorically defined) therefore satisfy the uniqueness condition.[166] Thus, this not only holds for arithmetic, but for *any* categorically definable structure (think of graph theory, or Euclidean geometry).

There are some things to be noted now that we have discussed the uniqueness condition. Remember that we have defined the domain of a substructure $\mathcal{M}_\varphi$ of $\varphi(\vec{x})$ to be the extension

159. Shapiro, *Philosophy of Mathematics*, 132.
160. Giovannini and Schiemer, 'What are Implicit Definitions?', 17.
161. Giovannini and Schiemer, 17.
162. The problem of reference is a hard one, and I will not say too much about it here. The model members of the class an implicit definition refers to may not be *known* to actually exist, but may be merely *possible* in the sense that they are isomorphic to the abstract structure that is defined.
163. Shapiro, *Philosophy of Mathematics*, 140.
164. Shapiro, 132.
165. Shapiro, 133.
166. Shapiro, 133.

of $\varphi(\vec{x})$ in that model. What happens when the defining theory is categorical (which it must be to structurally define a predicate according to the uniqueness condition), is that any two models of $\mathbb{T}$ are isomorphic, and there exists a bijection between the two domains of model $\mathcal{M}$ and $\mathcal{N}$. Any *submodel* for $\varphi(\vec{x})$ will make use of a subset of the domain of its parent model: $\{\vec{d} : \mathcal{M}_\varphi \vDash \varphi(\vec{d})\} \subseteq \mathcal{D}^{\mathcal{M}}$. Moreover, we said that a submodel $\mathcal{M}_\varphi$ interprets any part of the language in the same way as $\mathcal{M}$: $\sigma^{\mathcal{M}_\varphi} = \sigma^{\mathcal{M}}$.

Then, for two models $\mathcal{M}$ and $\mathcal{N}$ of a categorical theory $\mathbb{T}$, the submodels $\mathcal{M}_\varphi$ and $\mathcal{N}_\varphi$ must also be isomorphic. As there exists a bijection from the domain of $\mathcal{M}$ and the domain of $\mathcal{N}$ by Definition 3, and the domain of a substructure of a model is a subset of the domain of its parent model, such that in contains only the objects necessary for interpreting $\varphi(\vec{x})$, there must also exist a bijection between the domain of $\mathcal{M}_\varphi$ and the domain of $\mathcal{N}_\varphi$. Moreover, from isomorphism it follows that this bijection applied to an interpretation σ^M yields $\sigma^{\mathcal{N}}$. Since the interpretation of some $\sigma \in \varphi(\vec{x})$ by $\mathcal{M}_\varphi$ equals the interpretation of σ by $\mathcal{M}$, it also follows that this bijection between the domain of $\mathcal{M}_\varphi$ and the domain of $\mathcal{N}_\varphi$ satisfies isomorphism. From this a theorem follows immediately:

Theorem 2 (Substructure isomorphism) *If a theory $\mathbb{T}$ structurally defines some predicate $\varphi(\vec{x})$, then for all $\mathcal{M} \vDash \mathbb{T}, \mathcal{N} \vDash \mathbb{T}$, $\mathcal{M}_\varphi \simeq \mathcal{N}_\varphi$.*

The uniqueness condition thus is a concept that hangs together nicely with categoricity: any categorical theory that structurally defines a model, *uniquely* defines a model. The existence condition is a bit harder, especially since there is no mathematical counterpart that is as clear as categoricity. We will, however, consider the property of *coherence*, which will serve as a formal condition that can be used.

Definition 16 (Coherence) *$\varphi(\vec{x})$ is a coherent sentence (or formula) in a second-order language iff there is a structure that satisfies $\varphi(\vec{x})$.*[167]

Coherence is tied closely to the existence of mathematical objects and the problem of reference to those objects. Since we assume structuralism, mathematical objects ultimately come down to being (places in) structures, that exist if there is a coherent axiomatization of them. If we find a coherent axiomatization of a structure, we are sure that the structure exists according to Definition 16.[168] However, the notion of coherence should be made clear first, to prevent us from running in circles.

Coherence can be grounded in a deductive or a semantic principle. For coherence in a deductive sense, we can use the principle of *deductive consistency.*

Definition 17 (Deductive consistency) *A set of formulae Γ is* deductively consistent *iff no contradiction can be derived from the axioms in Γ.*[169]

A structural definition must then be consistent in the sense that no contradiction may be derived from the axioms. The semantic alternative is *satisfiability.* We will first give the definition of satisfiability:

Definition 18 (Satisfiability) *A set of formulae Γ is satisfiable iff there exists a model $\mathcal{M}$ and assignment function α, such that $\mathcal{M}, \alpha \vDash \varphi$ for all $\varphi \in \Gamma$.*[170]

167. Shapiro, *Philosophy of Mathematics*, 133.
168. Shapiro, 134.
169. Shapiro, 134.
170. Sider, *Logic for Philosophy*, 134.

In first-order logic, the two possible groundings for coherence coincide. First-order logic is complete, so any set of formulae is consistent if and only if it is satisfiable. In second-order logic, however, this is not the case: We can construct a set that is not satisfiable (and intuitively false according to the standard model) but consistent. Take Gödel sentence G. G might state the consistency of Peano arithmetic, and is true in second-order Peano arithmetic.[171] Also take set P, the axioms of second-order Peano arithmetic. We know that P is consistent: Would that not be the case, then a contradiction could be derived, from which *anything* would be provable, also the Gödel sentence for Peano arithmetic. We know that the Gödel sentence cannot be proved, nor can it be refuted. The Peano axioms thus must be consistent. The set of formulae $P \cup \{\neg G\}$ will still be consistent, as G is not provable, so no contradiction can be derived. However, $P \cup \{\neg G\}$ is not satisfiable, since Peano arithmetic *is* consistent, and the statement therefore is false. There is no model that is characterized by the set $P \cup \{\neg G\}$ (as any model is isomorphic to the standard model), so it is not coherent. Here coherence and deductive consistency part ways. However, satisfiability still can be a good way to model coherence.

There is a problem here, though: A model is a structure itself. To define coherence as satisfiability would be to ground existence of a structure in terms of existence of a structure; that will not work. However, being a model $\mathcal{M}$ comes down to being a member of the set-theoretic hierarchy, yet another structure.[172] After all, a model is nothing more than an ordered set consisting of a set of elements (the domain) and an interpretation function. For a formula to be satisfiable, there needs to exist a model, so there needs to exist a set. The coherence of set-theory therefore is crucial to have a working account of coherence based on satisfiability. If we have enough reason to accept set theory as a grounded framework, we can build our notion of coherence on it.

Shapiro mentions some reasons to believe we could accept set theory as a solid base. We cannot define mathematics (of which set theory arguably is part). It is not without reason that we chose not to work with the foundationalist conception of logic in Part 1: Grounding mathematics in a more precise and secure framework than mathematics never succeeded, and there are few reasons to think that it will ever work.[173] In first-order logic, both the completeness and incompleteness theorems hold. Part of the incompleteness theorems is also the mathematical fact that no theory can prove its own consistency.[174] However, it is very common to assume the consistency of the ZFC-axiomatization of set theory: No contradiction has been found yet and all things we intuitively want to be true are provable in ZFC.[175] Thanks to completeness in first-order logic, this means that there would exist a model, hence a set, that makes all of set theory true. This is, however, not the case for second-order logic: As there is no completeness in second-order logic, we cannot derive that there is a model of set theory by arguing for its consistency.

In second-order logic, Gödels completeness theorem does not hold. That means that, in principle, we could prove that there exists a model of ZFC: It would from there not follow that ZFC is consistent, thus no contradiction with Gödel's theorems is derived. Up to now, though, mathematicians have not succeeded in doing so. It is plausible that there exist (second-order) models of ZFC: The von Neumann universe is thought to be a good model of ZFC. Intuitively, it seems to model the axioms of ZFC in a good way, and it also aligns with how we usually think sets work. Again, this is not proved yet, but up to now there have been found no incongruities between the von Neumann universe and the axioms of ZFC. As a working hypothesis, it is thus assumed that the von Neumann universe models ZFC, hence set theory.[176] For a more in-depth and complete overview of the arguments why

171. Shapiro, *Philosophy of Mathematics*, 135.
172. Shapiro, 135.
173. Shapiro, 135.
174. Shapiro, 135.
175. Shapiro, 136.
176. Schoenfield, 'Axioms of Set Theory', 344.

the von Neumann universe can be thought of as modelling ZFC, please refer to Schoenfield, 1977.[177]

Due to the lack of proof that set theory has a working model, we cannot *define* coherence as satisfiability. Luckily, this is not necessary either. We could turn towards a somewhat more instrumentalist stance: what can we use as a formal, mathematical model for the intuitive notion of coherence? It turns out that satisfiability could well serve us.[178]

If we take *coherence* to be an intuitive notion, which cannot be explicated more, then the decision whether a theory is coherent also becomes a matter of intuition. However, there clearly is a link between satisfiability and coherence: Satisfiable theories denote a structure and theories that do not denote a structure are not satisfiable.[179] This also matches the *semantic* reading of Hilbert's views on structural definitions.[180] We could thus use satisfiability not as a definition, but as a formal model of our intuitive notion of coherence. In mathematical practice, the coherence of set theory is presupposed by many mathematicians: Both model theory and logic use set theory as their background ontology. Following this practice, we can rather safely state that satisfiability sufficiently models coherence.[181]

With satisfiability as an acceptable model of coherence, we can finally give the following two conditions for uniqueness and existence:

Existence A structural definition successfully characterizes a structure iff the theory that forms the definition is coherent, which we model by satisfiability.

Uniqueness A structural definition uniquely characterizes a structure iff the theory that forms the definition is categorical.

6 Structural Definitions and Meaning

Now we have seen what is a common theory of *how* formulae get their meaning according to structuralists, we are still left with the question *what* this meaning is. According to Ketland, the essence of an abstract structure eventually comes down to the propositional content of the formula that characterizes it.[182] What is this propositional content? The meaning of a formula in mathematical contexts is tied down by categorically characterizing the abstract structure, so what is the meaning that is given to the formulae that way? In this chapter, we will answer that question in both an informal and a formal way. The brief, informal answer will make use of the concepts discussed in the first part of the thesis, whereas the formal answer will make use of propositional functions, which will be discussed in terms of the concepts of the second part.

6.1 Informal Meaning: Structure and Formulae

As we have seen in the first part of this thesis, by abstracting from natural languages towards formal languages, we still ascribe a certain *intuitive meaning* to the resulting formulae. We will give a very brief account of how this intuitive meaning could be established.

The formula $\forall x(0 \neq S(x))$ intuitively states that zero is not the successor of any number. 'Zero' here means, as the places-as-object position states, the first position in any structure

177. Schoenfield, 'Axioms of Set Theory'.
178. Shapiro, *Philosophy of Mathematics*, 135.
179. Shapiro, 136.
180. Giovannini and Schiemer, 'What are Implicit Definitions?', 9.
181. Shapiro, *Philosophy of Mathematics*, 136.
182. Ketland, 'Abstract Structure', 29.

of arithmetic, not any concrete instantiation. There are two relevant ways, as discussed in the first part, in which the formalization of the natural language sentence can be done: by indifference to particulars and by de-semantification. We will only discuss the notion of meaning in a formal language that is formal as indifference to particulars here.

When we interpret the step of formalization as abstracting away from concrete objects, we incorporate the idea of places-as-objects: instead of concrete objects ('zero' as the abstract object zero), we give 'zero' the meaning of a place in the structure of the natural numbers. This step can already be taken before the step of abstraction and formalization: In natural language, the meaning of 'zero' could shift from a concrete object to a structure. However, the real step of abstraction where the abstraction is *made clear*, is in the step of formalization. As 0 formally has no meaning yet (it will be given a meaning by the semantics), we are left with the intended reading. Abstraction from individuals is one of the *goals* of formalization, so our intended reading should account for that.

Such an intended reading interprets the logical vocabulary in the usual way, but 0 should not be interpreted as a constant for any particular object. Less than particular objects, we aim for an interpretation that is *indifferent to particulars*, thus structural in meaning. Then 0 must be interpreted as an intuitive isomorphism class: 'the first place in the natural number structure'. The whole sentence $\forall x(0 \neq S(x))$ then *intuitively* has a meaning such that the formula is *true* for and in any model of arithmetic. The meaning of such a sentence thus should contain an element that, when applied to a structure, returns true or false. The meaning of a formula then consists of an element that yields a truth-value: It is true iff zero is not the successor of any number in the structure, and it will return false in any other case. We will formalize this idea using *propositional functions*.

6.2 Substructures and Propositional Functions

What is the element in a formula that returns true or false relative to a given structure? How is this element related to meaning? We will answer these questions in this section. We will argue for *propositional formulae* that return 'true' given a structure that satisfies it. Then, in the next section, we will give an example of how such formulae would work for both graph-theory and arithmetic.

Consider again predicate $\varphi_N(x)$: 'is a natural number'. Because it is easier to see how propositional formulae work in case of a bit lower-level predicate, we choose predicate $\varphi_P(x)$: 'is a natural number system' without loss of generality. $\varphi_P(x)$ is a *complex predicate* that is structurally defined by the theory of Peano arithmetic. The meaning of $\varphi_P(x)$ should be such that, applied to a structure, it is true if and only if that structure satisfies the theorems of Peano arithmetic.

A *propositional function* can account for this. Let us say that for $\varphi_P(x)$ the propositional formula is called Q. It is a function from structures to truth-values: $Q : S \to \{0, 1\}$. It must of course be defined when Q returns 0 or 1. Such propositional function may resemble the axioms of the theory, but there is an important difference: the relevant non-logical vocabulary is replaced by variables. Call this formula Φ_Q. So, if we know that a certain formula can determine whether Q holds in a model, Φ_Q should be independent of such a model. It should have variables for domains, constants and predicates. The assignment specification of the propositional function therefore will contain second-order variables. We can define this using logical terminology: For a structure S, the function returns 1 iff Φ_Q is true when the variables are assigned to the right vocabulary of S.

This directly relates to the meaning of a formula $\varphi(\vec{x})$. If we consider $\varphi_P(x)$, its meaning is not just this propositional function. It is the *class of structures* that make the propositional function true. This is independent of syntax, and two formulae that express the same propositional function eventually have the same meaning. This function on its own does not give us much information: an extra step is required in order to determine whether a certain structure lines up with the meaning of the formula. However, when we consider the meaning of the formula to be the isomorphism class of structures that make this propositional function true, this relation is immediately clear.

Theorem 3 *The meaning of any formula $\varphi(\vec{x})$ that is structurally defined by a theory $\mathbb{T}$ is fixed by $\mathbb{T}$.*

This immediately follows from categoricity. The meaning of any formula $\varphi(\vec{x})$ that is structurally defined by $\mathbb{T}$ is the isomorphism class of structures that satisfy $\varphi(\vec{x})$'s propositional function. As a theory has to be categorical to structurally define predicates, any model two models $\mathcal{M}$ and $\mathcal{N}$ of $\mathbb{T}$ are isomorphic. Therefore, any substructure $\mathcal{S}$ for $\varphi(\vec{x})$ must also be isomorphic to any other substructure for $\varphi(\vec{x})$ as follows from Theorem 2. Hence, all substructures for $\varphi(\vec{x})$ are contained in the isomorphism class of structures that satisfy the propositional function of $\varphi(\vec{x})$. That means that its meaning is fixed by the categorical theory: There is no meaning other than the submodels of $\mathbb{T}$ give us. Hence, the meaning of a formula $\varphi(\vec{x})$ in $\mathbb{T}$ is fixed by $\mathbb{T}$.

We can also say some things about the meaning of formulae that are not structurally defined by a *categorical* theory. There may be formulae in the language that have no meaning in the sense of a single isomorphism class. Take for example the theory of *first-order* Peano arithmetic. We know that this theory allows for *non-standard* models: There are models that make the theory true, but have a different structure compared to the intended model. A predicate defined by the first-order Peano axioms, $\varphi_{N1}(x)$, would then not just have one isomorphism class of models as its meaning.

When there are many non-isomorphic models that make the theory (and so formula $\varphi_{N1}(x)$) true, there are many isomorphism *classes* of models as well. That means that the meaning of $\varphi_{N1}(x)$ is not an isomorphism class with all models $N \simeq \mathcal{M}_{\varphi_{N1}}$, because there are many such models $\mathcal{M}_{\varphi_{N1}}$ that make $\varphi_{N1}(x)$ true, but are not isomorphic. Therefore, there also exist many isomorphism classes of models. That does not mean that $\varphi_{N1}(x)$ has no meaning. It does mean, however, that we cannot determine its meaning in an elegant way like we could in the case of a categorical theory.

In a sense, we can state that $\varphi_{N1}(x)$ has *many* meanings because of its many isomorphism classes. A more comprehensive way to put it, may be the statement that $\varphi_{N1}(x)$ is *ambiguous*. There exists more than one meaning of the formula, and all meanings are in a sense equally plausible: they all make the formula true. However, we still have the notion of an *intended model*: The model that interprets $\varphi_{N1}(x)$ in the way that we would intuitively call right. Formally, there is no way to distinguish between the different meanings. Hence, in case of a formula that is not structurally defined because it does not characterize a single structure, we can only say that its meaning is ambiguous.

If a formula $\psi(\vec{x})$ is nor defined by a categorical theory, nor coherent, it does not single out a structure. That means that there does not exist a model $\mathcal{M}_\psi \vDash \psi(\vec{x})$. So there also does not exist an isomorphism class of models: It is an empty class. Hence, a formula that is incoherent (for example due to a contradiction) has no meaning. At first sight, this might feel counter-intuitive. Why would a formula $(X(n) \wedge \neg X(n))$ have no meaning? When there are no models in which the formula is true, there also exist no objects a model could use to interpret the formula. From a structural realist perspective, this makes sense: If a formula cannot be made true, what would it refer to? A nonsensical formula does not refer to anything and in that sense has no meaning, just like a nonsensical sentence 'the ball is both round and square' does not have a meaning in that sense. Because of our choice for a model-theoretic semantics, reference determines the meaning and in that paradigm the idea of a formula not having a meaning fits well.

Summarizing, the *meaning of a formula* in a (mathematical) structural context is the isomorphism class of structures that satisfy its *propositional function.* A rough sketch of what such a function would look like is given above, but we will now give a more concrete application of this idea for the natural numbers. The account of meaning as an isomorphism class of structures is, however, not limited to graph or natural number structures. Any non-algebraic field of mathematics can be given such a theory of meaning. As long as the theory for such a field is able to *structurally define* formulae, it fixes the meaning of these formulae.

6.3 Propositional Functions in Arithmetic

Now that we have a general theory of meaning for structurally defined formulae, we can give an example of the meaning of predicate $\varphi_N(x)$: 'is a natural number'. The predicate $\varphi_N(x)$ is structurally defined by the Peano axioms. From these axioms, we can infer what is needed for a structure to be a natural number structure, and by checking whether an element is member of one of the structures of the right kind, we can say whether it is a natural number. The predicate $\varphi_N(x)$ can thus be split up: for some model $\mathcal{M}$ and element a, $\mathcal{M} \vDash N(a)$ iff $a \in X \wedge \varphi_P(X)$ with $\varphi_P(X)$: 'is a natural number structure'. X is a second-order variable and can be assigned a set in an interpretation.

$\varphi_P(X)$ is also a complex predicate, which we can split into several other (complex) predicates. We will use PA_2 here as an abbreviation for the second-order Peano axioms. The predicate $Def(X)$ stands for: 'the property of being a natural number can be defined in structure X'. Then $\varphi_P(X)$ stands for:

$$\mathcal{M}, \alpha \vDash S(X) \text{ iff } \mathcal{M}, \alpha \vDash PA_2 \wedge Def(X)$$

$$\mathcal{M}, \alpha \vDash Def(X) \text{ iff } X = \mathbb{N} \text{ iff } \mathcal{M}, \alpha \vDash \forall Y \subseteq X(z \in Y \wedge \forall n(n \in Y \to F(n) \in Y) \to Y = X)$$

So, the predicate $Def(X)$ is true in a model iff X equals the natural numbers, which we express using a second-order formula. We see that this formula contains three variables: X, F, z. X must be a set, the domain of the structure, F must be a function, the successor function, and z must be the system's element that denotes 0. A *structure of the right kind* can make this formula true or false: $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$ is such a structure consisting of a domain, function and interpretation of zero.

The meaning of a formula is the isomorphism set of structures that make its propositional function true. The propositional function of predicate $\varphi_N(x)$ is a function $P : (d, S) \to \{0, 1\}$ with d an object and S a structure of the right kind, so of the form $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$. The propositional function of PA_2 returns 'true' iff the axioms of second-order Peano arithmetic hold in the structure in question.

$$P(d, S) = 1 \text{ iff } \mathcal{M}, \alpha(a \mapsto d, X_1 \mapsto \mathcal{D}^{\mathcal{M}}, X_2 \mapsto F^{\mathcal{M}}, y \mapsto 0^{\mathcal{M}} \vDash a \in X_1 \wedge$$

$$\forall Y \subseteq X_1(0 \in Y \wedge \forall n(n \in Y \to X_2(n) \in Y) \to Y = X_1) \wedge PA_2$$

We see that any member of a system that instantiates the natural number structure makes the predicate N true. It has an interpretation of 0, a domain and a successor function such that the propositional function P is true: Def makes sure that there do not exist non-standard models and the Peano axioms make sure that for any number there is a successor (plus the other constraints). The meaning of $N(a)$ is thus given by its propositional function, that is true if a is a member of a valid natural number system.

Any member of a *non-standard model* of arithmetic will make the propositional formula P false: Def will not hold for such a structure. Also any system that fails to model all numbers will make P false as it does not satisfy the Peano axioms.

We have now seen an example of how a propositional function determines whether a certain structure satisfies a formula. The *meaning* of the formula is the class of all structures isomorphic to such a satisfying structure. Here, it would be the class of all structures that are such that $P(d, S) = 1$. All structures that make the propositional function true are in this case isomorphic to the natural numbers.

The *structure of the right kind*, which is here $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$, is exactly a φ-structure as described earlier. The model $\mathcal{M}_\varphi$ for formula $\varphi_N(x)$ that is defined by theory $\mathbb{T}_P$ of Peano arithmetic interprets exactly the non-logical vocabulary in $\varphi_N(x)$. $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$ are already part of any model $\mathcal{M}$ for $\mathbb{T}_P$, and as the structure $\mathcal{M}_\varphi$ is a substructure of $\mathcal{M}$, it will interpret $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$ in the same way as $\mathcal{M}$. What this means, is that any interpretation of the theory of Peano arithmetic yields an interpretation of $\varphi_N(x)$ by a substructure $\mathcal{M}_\varphi$. Hence, any such structure will be part of the isomorphism class of structures satisfying $\varphi_N(x)$'s propositional formula. Those structures will be of the form $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$ with the interpretation inherited from their superstructure.

$\varphi_N(x)$ will thus have as its meaning the isomorphism class of structures that make its propositional formula true. In this case that means the isomorphism class of $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$. Different models will interpret $\varphi_N(x)$ in a different way, but if their submodel's interpretation is isomorphic to $(\mathcal{D}^{\mathcal{M}}, F^{\mathcal{M}}, 0^{\mathcal{M}})$, it is a member of the isomorphism class and thus still a model for $\varphi_N(x)$. It is easy to see how this would work for a different nonalgebraic field of mathematics, for example graph theory. The propositional function returns true for a structure that satisfies the formula describing the abstract graph, and the meaning of that formula is the class of all structures isomorphic to a satisfying structure.

Conclusion

In this thesis, we have answered the question: *what is the meaning of a second-order formula in a structural, mathematical context?* We argued that the meaning of a second-order formula in such a context equals the isomorphism class of structures that satisfy its propositional function.

We have done this by laying out the basics of formal and logical languages, and discussing the formalization of mathematics in logic. Then, we have looked at Shapiro's account for second-order logic. We have seen that second-order logic is a very useful way of codifying mathematics, from a structuralist perspective. The fact that second-order logic can categorically define theories makes that structures can be uniquely determined. We have studied the meaning of mathematical statements that are made in second-order logic in the second part of the thesis.

First, we have laid out the differences between proof-theoretic and truth-conditional semantics and we have argued that truth-conditional semantics is the right choice from our perspective. We have also given the definition of full second-order semantics, in which we have done the rest of our work.

We then discussed an account of *structural definitions* that is informally given by Giovannini and Schiemer. We have formalized their account in terms of substructures, so that we could use it for answering the research question. Following the discussion regarding implicit and structural definitions, we used the results from there to give a formal account of the meaning of a formula in a structural context in terms of isomorphism classes. We have argued that the meaning of a mathematical statement from the viewpoint of structuralism is the isomorphism class of structures that satisfy the propositional function of a formula.

We have restricted our study to what we called *nonalgebraic fields of mathematics*. It would be natural to follow this research up by extending the results to algebraic fields of mathematics, that is, fields with mathematical theories that define concepts that determine more than one isomorphism class of structures. We leave this to further research.

Bibliography

Awodey, Steve, and Erich H. Reck. 'Completeness and Categoricity, Part II: Twentieth-Century Metalogic to Twenty-first-Century Semantics'. *History and Philosophy of Logic* 23, no. 2 (2002): 77–94. http://www.tandfonline.com/doi/abs/10.1080/0144534021000028619.

———. 'Completeness and Categoricity. Part I: Nineteenth-century Axiomatics to Twentieth-century Metalogic'. *History and Philosophy of Logic* 23, no. 1 (2002): 1–30. http://www.tandfonline.com/doi/abs/10.1080/01445340210146889.

Benacerraf, Paul. 'What Numbers Could not Be'. *The Philosophical Review* 74, no. 1 (1965): 47–73. https://www.jstor.org/stable/2183530.

Boolos, George S, John P Burgess and Richard C Jeffrey. *Computability and Logic*. 5th ed. 366. 2007.

Bourbaki, Nicolas. 'Foundations of Mathematics for the Working Mathematician'. *Journal of Symbolic Logic* 14, no. 1 (1949): 1–8. https://www.cambridge.org/core/product/identifier/S0022481200073102/type/journal_article.

Corcoran, John. 'Categoricity'. *History and Philosophy of Logic* 1, no. 1 (1980): 187–207. http://www.tandfonline.com/doi/abs/10.1080/01445348008837010.

Dedekind, Richard. *Was sind und was sollen die Zahlen?* Vieweg+Teubner Verlag, 1961.

Dummett, Michael. *Frege: Philosophy of Mathematics*. Cambridge, Mass.: Harvard University Press, 1995.

———. 'What Is A Theory of Meaning? (I)'. In *The Seas of Language*, 1–33. Oxford : New York: Clarendon Press ; Oxford University Press, 1993.

———. 'What Is A Theory of Meaning? (II)'. In *The Seas of Language*, 34–93. Oxford : New York: Clarendon Press ; Oxford University Press, 1993.

Dutilh Novaes, Catarina. 'The Different Ways in which Logic is (said to be) Formal'. *History and Philosophy of Logic* 32, no. 4 (2011): 303–332. http://www.tandfonline.com/doi/abs/10.1080/01445340.2011.555505.

Enayat, Ali, and Roman Kossak, eds. *Nonstandard Models of Arithmetic and Set Theory*. Contemporary Mathematics 361. Providence, R.I: American Mathematical Society, 2004.

Giovannini, Eduardo N., and Georg Schiemer. 'What are Implicit Definitions?' *Erkenntnis*, 2019. http://link.springer.com/10.1007/s10670-019-00176-5.

Gödel, Kurt. 'Die Vollständigkeit der Axiome des logischen Funktionenkalküls'. *Monatshefte für Mathematik und Physik* 37, no. 1 (1930): 349–360. https://doi.org/10.1007/BF01696781.

———. 'Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I'. *Monatshefte für Mathematik und Physik* 38, no. 1 (1931): 173–198. https://doi.org/10.1007/BF01700692.

Halbeisen, Lorenz, and Regula Krapf. *Gödel's Theorems and Zermelo's Axioms: A Firm Foundation of Mathematics*. Cham: Birkhäusert, 2020.

Hellman, Geoffrey. *Mathematics without Numbers*. Oxford University Press, 1993.

Horsten, Leon. 'Philosophy of Mathematics'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2019. https://plato.stanford.edu/archives/spr2019/entries/philosophy-mathematics/.

Iemhoff, Rosalie. 'Intuitionism in the Philosophy of Mathematics'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2020. https://plato.stanford.edu/archives/fall2020/entries/intuitionism/.

Kennedy, Juliette. 'Kurt Gödel'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2020. https://plato.stanford.edu/archives/win2020/entries/goedel/.

Ketland, Jeff. 'Abstract Structure'. Oxford, 10 January 2015.

Korbmacher, Johannes, and Georg Schiemer. 'What Are Structural Properties?†'. *Philosophia Mathematica* 26, no. 3 (2018): 295–323. https://academic.oup.com/philmat/article/26/3/295/3895509.

Kügler, Peter. 'Putting Brandom on His Feet: A Realist Interpretation of Inferentialism'. *Contemporary Pragmatism* 11, no. 2 (2014): 79–93. https://brill.com/view/journals/copr/11/2/article-p79_6.xml.

Leitgeb, Hannes. 'On Non-Eliminative Structuralism. Unlabeled Graphs as a Case Study, Part A'. *Philosophia Mathematica* 28, no. 3 (2020): 317–346. https://academic.oup.com/philmat/article/28/3/317/5805410.

Montague, Richard. *Formal Philosophy: Selected Papers of Richard Montague*. Edited by Richmond H. Thomason. Yale University Press, 1974.

———. 'Set Theory and Higher-Order Logic'. In *Studies in Logic and the Foundations of Mathematics*, 40:131–148. Elsevier, 1965.

Reck, Erich H. 'Dedekind's Structuralism: An Interpretation and Partial Defense'. *Synthese* 137, no. 3 (2003): 369–419. http://link.springer.com/10.1023/B:SYNT.0000004903.11236.91.

Reck, Erich H., and Georg Schiemer. 'Structuralism in the Philosophy of Mathematics'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2020. https://plato.stanford.edu/archives/spr2020/entries/structuralism-mathematics/.

Resnik, Michael D. 'Mathematics as a Science of Patterns: Ontology and Reference'. *Noûs* 15, no. 4 (1981): 529–550. https://www.jstor.org/stable/2214851?origin=crossref.

Rosser, J. Barkley. 'Review of Foundations of Mathematics for the Working Mathematician'. In collaboration with N. Bourbaki. *The Journal of Symbolic Logic* 14, no. 4 (1950): 258–259. https://www.jstor.org/stable/2269270.

Russell, Bertrand. *My Philosophical Development.* Unwin Books, 1975.

Schoenfield, J. R. 'Axioms of Set Theory'. In *Handbook of Mathematical Logic*, edited by Jon Barwise and H. Jerome Keisler, 321–344. Studies in Logic and the Foundations of Mathematics. Amsterdam ; New York: North-Holland Pub. Co, 1977.

Schroeder-Heister, Peter. 'Proof-Theoretic Semantics'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2018. https://plato.stanford.edu/archives/spr2018/entries/proof-theoretic-semantics/.

Shapiro, Stewart. *Foundations without Foundationalism: A Case for Second-order Logic.* Oxford logic guides 17. Oxford: Clarendon Press, 2000.

———. *Philosophy of Mathematics: Structure and Ontology.* New York: Oxford University Press, 1997.

Shapiro, Stewart, and Teresa Kouri Kissel. 'Classical Logic'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2018. https://plato.stanford.edu/archives/spr2018/entries/logic-classical/.

Sider, Theodore. *Logic for Philosophy.* Illustrated edition. Oxford ; New York: Oxford University Press, 2010.

Sipser, Michael. *Introduction to the Theory of Computation.* 3rd Ed. Boston, MA: Course Technology Cengage Learning, 2012.

Skolem, Thoralf. 'Über die Nicht-charakterisierbarkeit der Zahlenreihe mittels endlich oder abzählbar unendlich vieler Aussagen mit ausschliesslich Zahlenvariablen'. *Fundamenta Mathematicae* 23 (1934): 150–161. https://www.impan.pl/en/publishing-house/journals-and-series/fundamenta-mathematicae/all/23/0/93125/uber-die-nicht-charakterisierbarkeit-der-zahlenreihe-mittels-endlich-oder-abzahlbar-unendlich-vieler-aussagen-mit-ausschliesslich-zahlenvariablen.

Sundholm, Göran. 'Proof Theory and Meaning'. In *Handbook of Philosophical Logic: Volume III: Alternatives in Classical Logic*, edited by D. Gabbay and F. Guenthner, 471–506. Synthese Library. Dordrecht: Springer Netherlands, 1986.

Tarski, Alfred, and John Corcoran. 'What Are Logical Notions?' *History and Philosophy of Logic* 7, no. 2 (1986): 143–154. http://www.tandfonline.com/doi/abs/10.1080/01445348608837096.

Tennant, Neil. 'Logicism and Neologicism'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2017. https://plato.stanford.edu/archives/win2017/entries/logicism/.

Väänänen, Jouko. 'Second-order and Higher-order Logic'. In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2020. https://plato.stanford.edu/archives/fall2020/entries/logic-higher-order/.

———. 'Second-Order Logic and Foundations of Mathematics'. *Bulletin of Symbolic Logic* 7, no. 4 (2001): 504–520. https://www.cambridge.org/core/product/identifier/S1079898600005370/type/journal_article.